Novel Wire Density Driven Full-Chip Routing for CMP Variation Control

Huang-Yu Chen[†], Szu-Jui Chou[†], Sheng-Lung Wang[§], and Yao-Wen Chang[†‡]
[†]Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan
[‡]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[§]Synopsys, Inc, Taipei, Taiwan

***Abstract*— As nanometer technology advances, the post-CMP dielectric thickness variation control becomes crucial for manufacturing closure. To improve CMP quality, dummy feature filling is typically performed by foundries after the routing stage. However, filling dummy features may greatly degrade the interconnect performance and lead to explosion of mask data. It is thus desirable to consider wire-density uniformity during routing to minimize the side effects from aggressive post-layout dummy filling. In this paper, we present a new full-chip grid-based routing system considering wire density for reticle planarization enhancement. To fully consider wire distribution, the router applies a novel two-pass, top-down planarity-driven routing framework, which employs a new density critical area analysis based on Voronoi diagrams and incorporates an intermediate stage of density-driven layer/track assignment based on incremental Delaunay triangulation. Experimental results show that our methods can achieve more balanced wire distribution than state-of-the-art works.**

I. Introduction

As IC process geometries shrink to 65nm and below, one important yield loss of interconnects comes from the *chemical-mechanical polishing (CMP)* step in the copper metallization (*Damascene*) process. Because of the difference in hardness between copper and dielectric materials, the CMP planarizing process might generate topography irregularities. A non-uniform feature density distribution on each layer causes CMP to over polish or under polish, generating metal dishing and dielectric erosion [22]. These thickness variations have to be carefully controlled, since the variation in one interconnect level is progressively transferred to subsequent levels during manufacturing, and finally the compounding variation can be significant on an upper level, which is often called the multi-layer accumulative effect [23].

Two key problems arise from the post-CMP thickness variation: (1) the layout surface fluctuates inside or outside the *depth of focus (DOF)* of the photolithography system, such that the exposed patterns do not appear acceptably sharp and open/short defects may even occur, and (2) these irregular variations greatly change the electrical characteristics of interconnects, especially for resistance and capacitance, degrading the accuracy of timing analysis and worsening the electromigration. As a result, in order to improve chip thickness uniformity, TSMC recommends performing virtual CMP (VCMP) analysis to identify the metal and dielectric thickness variation hotspot before chip fabrication for 65nm manufacturing processes (see TSMC Reference Flows 7.0) [24].

In order to improve the CMP quality, modern foundries often impose recommended layout density rules and fill *dummy features* into layouts to restrict the variations on each layer. Dummy features may either be connected to power/ground *(tied fills)* or left floating *(floating fills)* [19]. The tied fill has predictable but higher capacitance, while the floating fill has lower but unpredictable one due to the floating nature. Traditionally, electrical impacts of dummy fills can be negligible, and dummy features are inserted during the post routing stage. Filling algorithms have been proposed to satisfy density bounds and reduce the density variation [16], [25]. However, as reported in [26], these filled dummy features may incur troublesome problems at 65nm and successive technology nodes. The tied fill may induce crosstalk for its high coupling capacitances to nearby interconnects and would place a heavy burden for P/G (power/ground) network. On the other hand, the floating capacitance of floating fills is usually uncertain, and thus the induced coupling capacitance might unpredictably harm the timing-optimized results in the previous design stages. Moreover, dummy fills also sheerly increase the data volume of mask, lengthening the time of mask-making processes such as mask synthesis, writing, and inspection verification. Especially, these filled features would significantly increase the input data in the following time-consuming reticle enhancement techniques, such as OPC (optical proximity correction) and PSM (phase shift mask). Therefore, much research focuses on impact-limited dummy feature filling algorithms [7], [18].

In the nanometer technology, routing has become a decisive factor for determining chip manufacturability, since it presides over most of the layout geometries in the back-end design process. In order to tackle these manufacturing challenges, routing techniques must handle the increasing complexity. The routing approaches applying the bottom-up *coarsening* and top-down *uncoarsening* techniques have demonstrated the superior capability of handling large-scale routing problems, such as the Λ-shaped multilevel [3], [4], [12], the V-shaped multilevel [5], and the two-pass bottom-up [6] routing frameworks.

Recently, routing considering wire distribution has attracted much attention in the literature. The earlier studies for CMP processes have indicated that the post-CMP dielectric thickness is highly correlated to the layout pattern density, because during the polishing step, interlevel dielectric (ILD) removal rates are varied with the pattern density [23]. Further, the layout pattern (consisting of wires and dummy features) density can be systematically determined by the wire density distribution, as reported in [9]. Therefore, managing wire density at the routing stage has great potential for alleviating the aggressive dummy feature filling induced problems.

Li *et al.* [20] presented the first routing system in the literature addressing the CMP induced variation. By setting the desired density in the cost function of global routing, the routing results have more balanced interconnect distribution. Cho *et al.* [9] proposed a pioneering work to consider CMP variation during *global* routing.

This work was supported in part by UMC and NSC of Taiwan under Grant No's. NSC 96-2752-E-002-008-PAE, NSC 96-2628-E-002-248-MY3, NSC 96-2628-E-002-249-MY3, and NSC 96-2221-E-002-245.

They empirically developed a predictive CMP density model and showed that the number of inserted dummy features can be predicted by the wire density. Therefore, they proposed a minimum-pin density global routing algorithm to reduce the maximum wire density in each global tile. However, both approaches only consider the wire density *inside* a routing tile. Since the topographic variation is a long-range effect, focusing density value inside each routing tile may incur larger inter-tile density difference and result in more irregular post-CMP thickness. (See Fig. 1 (a).) Therefore, optimizing wire-density uniformity inside a routing tile is obviously not a right metric and a common pitfall for CMP control. For better CMP control, it is more desirable to minimize the global variation of wire density, i.e., the density gradient. As the example shown in Fig. 1, if the density lower and upper bounds are 20% and 80% respectively, then the three adjacent routing tiles in Fig. 1 (b) all satisfy these rules. However, Fig. 1 (c) is a better choice for CMP control because it has the minimum wire-density gradient.

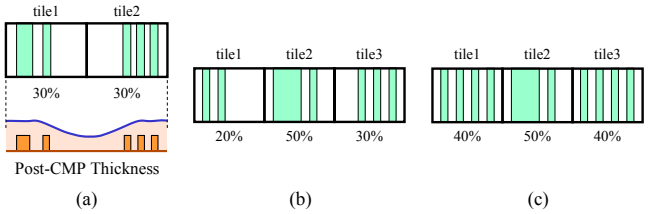


Fig. 1. Density variation among neighboring subregions impacts topography. (a) Different wire distribution in a subregion exists even under the same density. Large density variation among neighboring subregions leads to post-CMP thickness irregularities. (b) Three adjacent routing tiles satisfy density rules but result in unbalanced wire distribution. (c) A better result for minimizing the density gradient among tiles.

In this paper, we present a new full-chip grid-based routing system, named TTR (Two-pass Top-down grid-based Router), considering wire-distribution uniformity for density variation minimization. To fully consider wire distribution, the router is based on a novel two-pass, top-down planarization-driven routing framework. (See Fig. 2 for an illustration.) Different from the aforementioned works, TTR has the following distinguished features:

- A new routing framework of performing density prediction in the *prerouting* stage, followed by planarization-aware *global* routing at the first uncoarsening stage, an intermediate stage of density-driven *layer/track assignment*, and then *detailed* routing at the second uncoarsening stage.
- An efficient density critical area analysis (CAA) algorithm based on *Voronoi diagrams* is performed *off-line* in the pre-routing stage, which considers *both* topological information of pins and wire connection to complement the density analysis. As shown in Section IV, the Voronoi-diagram based CAA algorithm leads to 3–5% faster overall routing process due to easier density control for later detailed routing. Further, it can substantially improve the resulting wire-density uniformity.
- A planarization-aware global router is employed to consider the density lower and upper bounds while minimizing the density *gradient* among global tiles.
- A layer assigner for panel-density minimization and a density-driven track assignment algorithm based on the incremental *Delaunay triangulation* are performed before detailed routing to preserve more flexibility for wire density arrangement.

Compared with the density-driven routing system [20], experimental results show that TTR can achieve 43% reduction on the maximum number of nets crossing in tiles and obtain at least 35% smaller standard deviations of wire distribution.

The rest of this paper is organized as follows. Section II describes the routing model and the routing framework. Section III presents our density-driven routing algorithms. Experimental results are reported in Section IV, and conclusions are given in Section V.

II. Routing Model

We first explain the routing model. As illustrated in Fig. 2, G_k corresponds to the routing graph of level k. Each level contains a number of *global cells* (GCs), and the GCs belonging to different levels have different sizes. We denote GC_k as the GC of level k.

The first top-down routing pass is for global routing, which starts uncoarsening from the coarsest level to the finest level (level 0). At each level k, our global router finds routing paths for the *local nets* (those nets that entirely sit inside GC_k but not inside GC_{k-1}). After all the global routings of level k are performed, we divide one GC_k into four smaller GC_{k-1} and at the same time perform resource estimation for use at level k-1. Uncoarsening continues until the size of GC_k at a level is below a threshold.

The second top-down routing pass is for detailed routing. As the first pass, it processes uncoarsening from the coarsest level to the finest level. At each level, a detailed router is performed and rip-up/re-route procedures are applied for failed nets. The process continues until we reach level 0 when the final routing solution is obtained.

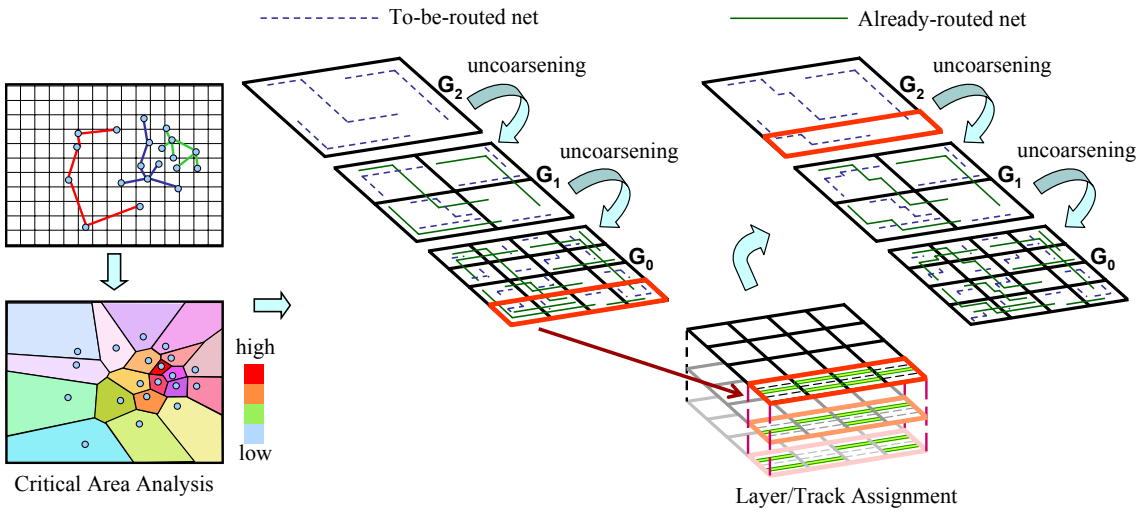
III. Density-Driven Routing

To deal with wire density optimization, we develop a Two-pass Top-down full-chip grid-based Routing system, named TTR (see Fig. 2). The rational for top-down routing lies in the fact that it tends to route longer nets first level by level, which directly contributes to better wire planning since longer nets have greater impacts on planarization than shorter ones. We detail the three distinguished stages of TTR in the following subsections.

A. Density Critical Area Analysis (CAA)

In order to guide the following routing for making better decisions, TTR features a density critical area analysis in the prerouting stage that identifies the potential over-dense hotspots. Recently, Cho *et al.* [9] performed *minimum-pin density* routing to prevent global-routing paths from crossing through over-dense areas. The reason is that a path with higher pin density tends to pass through more wire dense areas, since the existence of a pin means that eventually there is at least one wire connecting to other pins. This approach can help reduce the wire density in each global tile. However, there are some limitations. As the global routing instance shown in Fig. 3 (a), although the routing path n_1 passes fewer pins, it may exacerbate the over-dense areas in its adjacent regions. In contrast, the routing path n_2 contains more pins but results in a better balanced wire distribution. Moreover, the pin density is not directly proportional to the wire density. As shown in Fig. 3 (b), the small pin count in the global tile may still contribute to large wire density.

Therefore, it is necessary to consider *both* topological information and wire connections of each pin to complement the density analysis. To remedy the deficiencies, we develop a new enhanced analysis model based on *Voronoi diagrams*. The Voronoi diagram of a point set P partitions the plane into regions, called *Voronoi cells*, each of which is associated with a point of P. If a point in the plane is closer to the point $p_t \in P$ than to any other point of P, then this point will be in the interior of the Voronoi cell associated with p_t. The boundary



Prerouting Stage
Identify the potential density hot spots based on the pin distribution and wire connection to guide the following global routing.

First Pass Stage
Apply prerouting-guided planarization-aware global pattern routing for local nets and iteratively refine the solution.

Intermediate Stage
Perform density-driven layer/track assignment for long segments panel by panel.

Second Pass Stage
Use segment-to-segment detailed maze routing to route short segments and reroute failed nets level by level.

Fig. 2. The new two-pass, top-down routing framework.

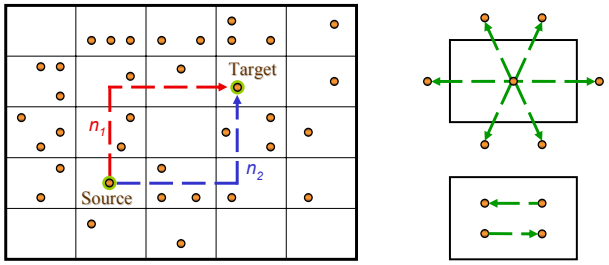


Fig. 3. Limitations of minimum-pin density routing [9]. (a) Path n_1 passes fewer pins but tends to exacerbate the over-dense areas in its adjacent regions, whereas path n_2 passes more pins but leads to better balanced wire density. (b) Pin count cannot reflect the wire density in the global tile well.

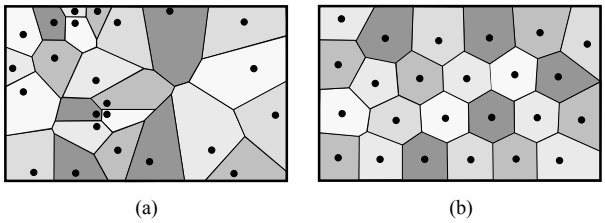


Fig. 4. Voronoi diagram for points with (a) non-uniform distribution and (b) uniform distribution.

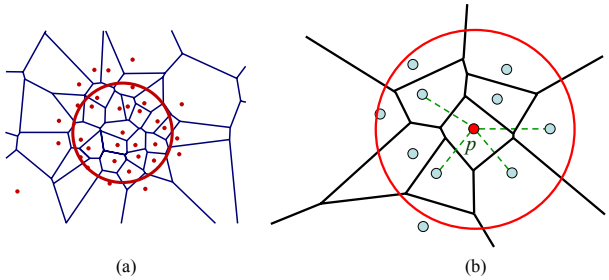


Fig. 5. Voronoi-diagram-based pin density analysis. (a) Proximity relation induced by the Voronoi diagram reflects the dense quantity well. (b) Density cost is measured by the topological proximity and the number of wire connections.

segments of a Voronoi cell are called the *Voronoi edges*. A Voronoi diagram can efficiently compute the physical proximity and has been well studied in computational geometry [13]. Papadopoulou and Lee [21] used Voronoi diagrams of rectilinear polygons to compute the critical areas for short defects in a circuit layout.

The motivation for the Voronoi diagram approach lies in the following observation.

Observation 1: Given the Voronoi diagram of points, the standard deviation for the size of Voronoi cells strongly depends on the distribution of these points.

As illustrated in Fig. 4 (a), the Voronoi cells for points with non-uniform distribution have large variation in sizes; in contrast, as shown in Fig. 4 (b), for points with uniform distribution, the sizes of Voronoi cells are almost the same.

Another observation can quantify the proximity relation to indicate whether a point lies in the dense area.

Observation 2: For a point, the number of adjacent Voronoi cells which entirely sit within a specified distance from this point reflects the dense quantity of the region where this point lies.

As shown in Fig. 5 (a), the point in the dense area has more Voronoi cells around it within a given circle with its center at this point.

Base on these observations, we specify a range r and associate each pin p with a density cost d_p, which is defined as

$$d_p = \alpha\nu_p + (1-\alpha)\omega_p, \tag{1}$$

where ν_p is the number of Voronoi cells around p (excluding the

Voronoi cell associated with p itself) which entirely sit inside the circle with a center at p and radius r, ω_p is the number of wire connecting to p, and $\alpha, 0 \leq \alpha \leq 1$, is a user-defined parameter. For the example shown in Fig. 5 (b), there are three Voronoi cells around p which entirely sit inside the circle, and four wires are connected to p. Therefore ν_p and ω_p equal 3 and 4, respectively.

In the current implementation, we set the radius r as the average distance among pins of adjacent Voronoi cells. In this way, the expected value for ν_p would be zero if p lies in a uniformly distributed region; otherwise, ν_p would increase as a penalty to reflect the density hotspot where p lies. Additionally, since two-pin nets practically dominate the netlist in most designs, the expected value of ω_p would equal one. Therefore, the ranges of ν_p and ω_p in Eq. (1) are similar and can be reasonably combined together through the α parameter.

After all density costs of pins have been computed, we transform these costs into the cost of global tiles. For each global tile t, we set its predicted density cost $\widetilde{d_t} = \max\{d_p \mid p \text{ is inside } t\}$ in the prerouting stage. Then TTR feeds the pre-estimated density information to the following routing stages. The density critical area analysis can be efficiently performed. We have the following theorem.

Theorem 1: The Voronoi-diagram based density CAA runs in $O(|P| \lg |P|)$ time, where $|P|$ is the number of pins.

Note that the Voronoi-diagram based CAA algorithm is performed only once, and its running time overhead is very small (about 3% of the total running time in our experiment). Further, it even leads to 3–5% faster overall routing process due to easier density control for later detailed routing, and it can substantially improve the resulting wire-density uniformity.

B. Planarization-Aware Global Routing

The global routing plans tile-to-tile routing paths for all nets and thereby is an important step to decide the wire distribution and maintain a uniform metal density across the chip. As mentioned in the introduction, both previous works [9], [20] consider only the wire density *inside* each global tile, which might incur larger inter-tile density gradient and thus more irregular post-CMP thickness. As a result, for better CMP control, a global router has to consider the density variation (gradient) among global tiles in addition to wire density inside each tile.

In our TTR, the global routing performed in the first top-down uncoarsening pass is based on pattern routing [17]. Pattern routing uses an L-shaped (1-bend) or Z-shaped (2-bend) route to make the connection, which gives the shortest path length between two points while reducing the routing bends. Therefore, the obtained routing path is the shortest, and we thus can focus on the objectives that we most concern.

We define the planarization-aware cost Φ_t for each global tile t as follows:

$$\Phi_t = \widetilde{d_t} + \begin{cases} \kappa_p, & \text{if } d_t \geq B_u \\ \beta(2^{d_t} - 1) + (1-\beta)(d_t - \overline{d_t})^2, & \text{if } B_l \leq d_t < B_u \\ \kappa_n, & \text{if } d_t < B_l \end{cases} \tag{2}$$

where d_t is the wire density of t, $\widetilde{d_t}$ is the predicted hotspot cost calculated in the prerouting stage, $\overline{d_t}$ is the average wire density of tiles adjacent to t, B_l and B_u are density lower and upper bounds specified in foundry density rules respectively, and $\beta, 0 \leq 1$, is a user-defined parameter. (Note that both the values of $2^{d_t} - 1$ and $(d_t - \overline{d_t})^2$ are between 0 and 1.) κ_p and κ_n are constants, where κ_p is a positive penalty that hinders the over denseness in the global tile, and κ_n is a negative reward that encourages paths to go through sparse tiles. The second equation simultaneously considers local density and minimizes the density difference among adjacent regions.

For more balanced wire distribution, the cost function Φ_p of the global routing path g_p is defined as follows:

$$\Phi_p = avg\{\Phi_t \mid \text{tile } t \text{ is on the path } g_p\}, \tag{3}$$

in which the average manner can represent the consciousness of even wire distribution.

C. Density-Driven Layer/Track Assignment

Recently, Cong *et al.* [11] proposed the first wire-planning scheme between global and detailed routers to reduce congestion. Batterywala *et al.* [2] also suggested to add a track assignment stage between global and detailed routing to improve the routing quality. Ho *et al.* [14] developed a layer/track assignment heuristic in the intermediate stage for crosstalk optimization. Later in [15], Ho *et al.* further extended their track assigner for the wirelength reduction in X-architecture routing. However, wire density is not addressed in these works.

1) Density-Driven Layer Assignment: In this paper, we propose a new layer/track assignment algorithm for wire-density optimization. To our best knowledge, this is the first work of wire planning that addresses the wire-density optimization in the literature.

We handle long horizontal (vertical) segments which span more than one complete global tile in a row (column) in the middle layer/track assignment stage and delegate short segments to the detailed router. The full row (or column) of a global tile array is called a *row (column) panel*. We will refer to a row panel as a panel throughout the paper for brevity, unless specified otherwise.

In a panel, the *local density* of a column is defined as the total number of segments and obstacles at that column, and the *panel density* is the maximum local density among all columns. For example, Fig. 6 (a) gives a row panel with 11 columns, c_1 to c_{11}. There are six segments s_1 to s_6 in the panel and two obstacles o_1 and o_2 in layers, and its panel density is equal to 4. We intend to evenly arrange these segments to two horizontal layers (say layers 1 and 3) while minimizing the panel density at each layer. The density-driven layer assignment problem is defined as follows.

- **The Density-driven Layer Assignment (DLA) Problem:** Given a set L of layers, a set S of disjoint segments in a panel, and a set O of fixed obstacles in layers, assign each segment of S to a layer, such that for each layer the local density is balanced, and the panel density is minimized.

To solve the DLA problem, we partition the segments and obstacles in each panel into $|L|$ layer groups such that the main objective of DLA is achieved.

First, we build the horizontal constraint graph $HCG(V, E)$ for S and O in the panel. Each vertex $v \in V$ corresponds to a segment or an obstacle, and two vertices v_i and v_j are connected by an edge $e \in E$ if their spans overlap. The cost of edge $e(v_i, v_j)$ is defined as the maximal local density among the overlapping columns between v_i and v_j. With this weighting policy, if two vertices are connected by an edge with a high cost, they should be separated into different layers. Fig. 6 (b) shows the HCG of the panel in Fig. 6 (a). Here, the obstacle o_2 and segment s_3 overlap in columns c_3 and c_4, and the maximal local density of c_3 and c_4 is 3. So the cost of the edge (o_2, s_3) equals 3.

Consequently, we can formulate the DLA problem as a max-cut, k-coloring problem (MCP) [10] on the HCG graph, where k equals $|L|$. In this way, we can guarantee that the partitioning result can evenly

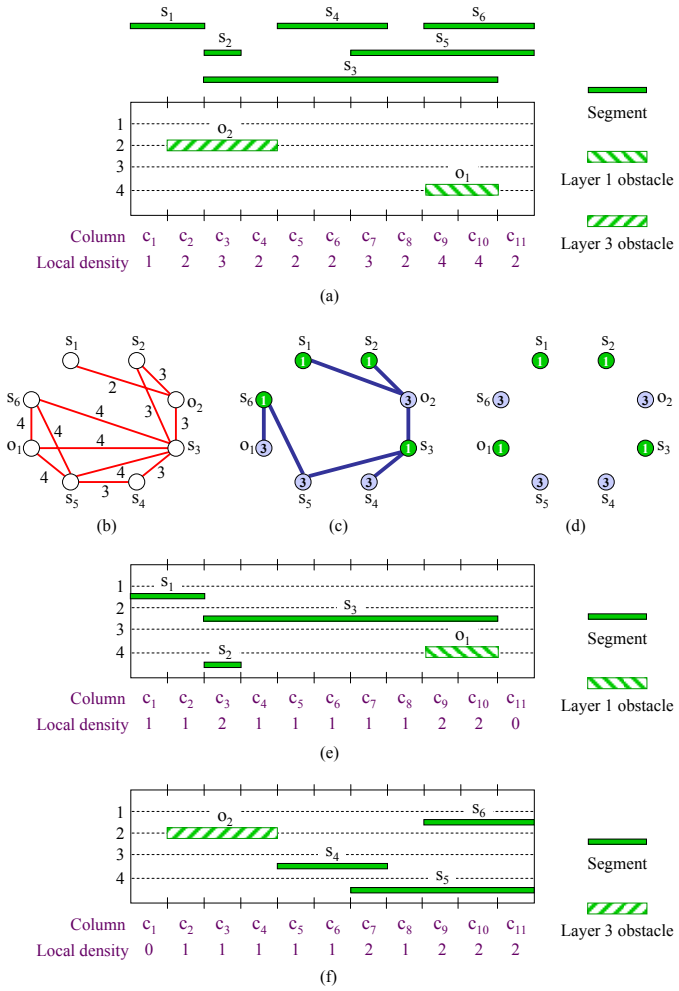


Fig. 6. A density-driven layer assignment example. (a) A row panel A consists of six segments and two obstacles. We intend to evenly assign these segments to two horizontal layers (layers 1 and 3). (b) The horizontal constraint graph. (c) The layer-partitioning result for two layer groups by applying the maximum spanning tree and k-coloring algorithms. (d) The final layer assignment result by applying a minimum-impact repair procedure to exchange the layers of s_6 and o_1. (e) and (f) The final local densities of layers 1 and 3, respectively.

distribute the segments of the maximal local density to different layer groups. However, the MCP is NP-complete [10]. Thus, we resort to a simple, yet efficient heuristic by constructing a maximum spanning tree on the HCG and applying a k-coloring algorithm on this tree. Note that the k-coloring algorithm on a *tree* can be solved in linear time. Fig. 6 (c) shows a layer-partitioning result of Fig. 6 (a), where s_1, s_2, s_3 and s_6 are partitioned as one layer group, and o_2, s_4, s_5 and o_1 are partitioned as another one. Note that the objects o_1, s_3, s_5, and s_6 at columns c_9 and c_{10} that induce the maximum local density are separated into two different layer groups.

At the last step, since obstacles are already in fixed layers, we applied a minimum-impact repair procedure for obstacles. If an obstacle is not placed in the right layer (e.g., o_1 in Fig. 6 (c)), the layer of a vertex v_o of an obstacle is exchanged with that of a vertex v_s of a segment such that the edge cost (v_o, v_s) is the maximum among the edges connected with v_o in the maximum spanning tree. If there does not exist such a vertex v_s, we can just assign v_o to the correct layer since there is no segment there (otherwise, there must be an edge connected with v_o). The final assignment result after the repair procedure for exchanging the layer of vertex o_1 with that of vertex s_6 is shown in Fig. 6 (d). As a result, the final assignment has a very balanced density distribution that the average local density of layer 1 is 1.18 and that of layer 3 is 1.27 while the panel densities in both layers equal 2. See Figs. 6 (e) and (f) for the resulting segment assignments for layers 1 and 3, respectively.

Note that for practical concern, in addition to the objectives of DLA, a good/practical layer assigner shall also assign layers with more segments of the same nets closer to each other to minimize the stacked-via usage. We can model the connectivity among layers as a connection graph $C(V, E)$ whose nodes represent layers and edges denote the corresponding connectivity. Then, the problem can be solved by first computing the Maximum-Weighted Hamiltonian Path (MWHP) on $C(V, E)$ and then assigning layers with the largest connectivity closer to each other. Since the MWHP problem is NP-hard, we apply a greedy algorithm similar to Kruskal's minimum spanning tree algorithm to handle the MWHP problem. We first sort edges by their weights, and then add edges in non-increasing weight order if they form a path.

2) Density-Driven Track Assignment: After the layer assignment, we intend to uniformly spread the segments in each layer of panels and balance the segment distribution among neighboring panels. For convenience, we hereafter refer to a layer of a panel as a *panel* since the layer assignment has already been performed. Let $\mathcal{T}$ be the set of tracks inside a panel. Each track $\tau \in \mathcal{T}$ can be represented by the set of its constituent contiguous intervals. Denoting these intervals by x_i. A segment $s \in S$ is said to be assignable to $\tau \in \mathcal{T}$, $\tau \equiv \biguplus x_i$, if either x_i is a free interval or is an interval occupied by a segment of the same net. The density-driven track assignment problem is defined as follows:

- **The Density-driven Track Assignment (DTA) Problem:** Given a panel A and its two neighboring panels A_u and A_b, a set of tracks $\mathcal{T} \in A$, a set of segments $S \in A$, and a set of fixed obstacles $O \in A$, for a given cost function $\Psi : S \times \mathcal{T} \rightarrow \mathbb{R}$ which represents the density cost of assigning a segment to a track, find a feasible assignment of S to $\mathcal{T}$ that minimizes Ψ.

To solve this problem, we propose an Incremental Delaunay-triangulation-based Track Assignment (IDTA) algorithm. In Observation 1, we have discovered the relation between density uniformity and the Voronoi diagram. Instead of using the Voronoi diagram, we can leverage the good properties of its dual graph, called *Delaunay Triangulation* (DT), to evaluate the segment distribution. The DT for a point set is a triangulation that minimizes the standard deviations of angles among all triangles, and the circumscribed circle of every triangle will not contain any other point in its interior [13]. Similar to the Voronoi diagram, the standard deviation for the size of triangles in DT can reflect the distribution of these points. Thus, we can represent each segment by three points, two end points and one center point, and analyze the corresponding DT of these points.

Before performing the IDTA algorithm, we first model the distribution of segments and obstacles in each neighboring panel into an *artificial segment* lying on the boundary of A. In order to reflect the distribution of objects in a neighboring panel A_n of A, we set the length of an artificial segment as the average occupied length per track in A_n, and the center of this artificial segment is determined by the center of gravity of all segments and obstacles in A_n.

Fig. 7 shows the IDTA algorithm. Without loss of generality, we discuss the track assignment at a row panel, and the case for a column panel is similar. For the track assignment problem, the x-coordinates

Algorithm: IDTA
Input: A /* The panel */
S /* A set of segments */
O /* A set of fixed obstacles */
s_u, s_b /* The artificial segments */
Output: T /* The assignment configuration */
1 **for** each segment $s_i \in S$
2 Compute the flexibility of s_i, $\xi(s_i)$;
3 $T \leftarrow \emptyset$;
4 Construct an initial point set P based on $O \cup \{s_u, s_b\}$;
5 Construct an initial DT of P;
6 **while** S is not empty
7 Choose the segment s_j with the smallest flexibility;
8 Determine $track(s_j)$ such that the maximum area difference among the introduced triangles is minimum;
9 $T \leftarrow T \cup \{s_j, track(s_j)\}$;
10 Add the points introduced by s_j into P;
11 Update DT incrementally;
12 $S \leftarrow S - \{s_j\}$;
13 **for** each $s_k \in S$ overlapping s_j
14 Update $\xi(s_k)$;
15 Return T;

Fig. 7. The Incremental Delaunay-triangulation-based Track Assignment (IDTA) algorithm.

of segments are fixed (i.e., the segments in row panels can only move in the vertical direction), so we can focus on the y direction. At the beginning, we define the *flexibility* of a segment s_i as

$$\xi(s_i) = t_i + \frac{1}{\ell_i},$$

where t_i is the number of assignable tracks of s_i, and ℓ_i is the length of s_i. Since the x-coordinate of s_i is fixed, t_i can easily be computed. If the flexibility of s_i is smaller, which means that s_i might have longer length or less space to insert, then s_i should be assigned first.

After the flexibility computation, we construct an initial DT that includes only the obstacles and two artificial segments. Each segment or obstacle is represented as three points, its left-end, center, and right-end points. Fig. 8 (a) shows the initial DT. The construction of DT takes $O(|P| \lg |P|)$ time, where $|P|$ is the number of points. Note that a DT can be updated incrementally; if a new point is added into an existing DT, we only need to update the triangles introduced by this new point. Therefore, the process can be performed very efficiently. The update will be frequently used in the following steps.

Lemma 1: Adding a new point into an existing Delaunay triangulation of $|P|$ points takes $O(\lg |P|)$ time.

Segments are assigned sequentially in the non-decreasing order of their flexibilities. Suppose segment s_j has the smallest flexibility among all unassigned segments, then we assign s_j to a proper track. In order to minimize the area difference among all triangles, the track which results in a DT with smaller area difference is preferred.

After assigning s_j to the track $track(s_j)$, we need to update the DT and the flexibility of segments. Since we can incrementally update the DT, only the new triangles introduced by s_j need to be re-generated. Only the segments that overlap s_j and are originally assignable to $track(s_j)$ need to update their values of flexibility. For those segments, the new flexibility would be the original flexibility minus 1. The number of segments overlapping with s_j is bounded by $\ell_j \times t_j$, which is bounded by the constant size of the panel; here, ℓ_j is a value, and t_j is bounded by the number of tracks in a panel, which is predetermined before the routing and is around 10–20 in our implementation. Therefore, the total time complexity of updating DT and the flexibilities of segments is $O(\lg |S|)$, and we have the following theorem for the overall time complexity of the IDTA algorithm.

Theorem 2: The IDTA algorithm runs in $O(|S| \lg |S|)$ time, where $|S|$ is the number of segments in a panel.

Fig. 8 shows a track assignment example. Fig. 8 (a) is the initial DT including only obstacles and artificial segments, and Figs. 8 (b), (c), (d) are the assignment results of s_3, s_2, and s_1, respectively. The flexibilities of unassigned segments are listed on the right side of the figures. Note that each time when a segment is assigned, the flexibilities of unassigned segments are incrementally updated.

After the track assignment, the actual track position of a segment is known. Thus, we can perform classical segment-to-segment maze routing in the detailed routing stage to connect shorter nets which span at most two routing tiles, and the whole routing process is finished.

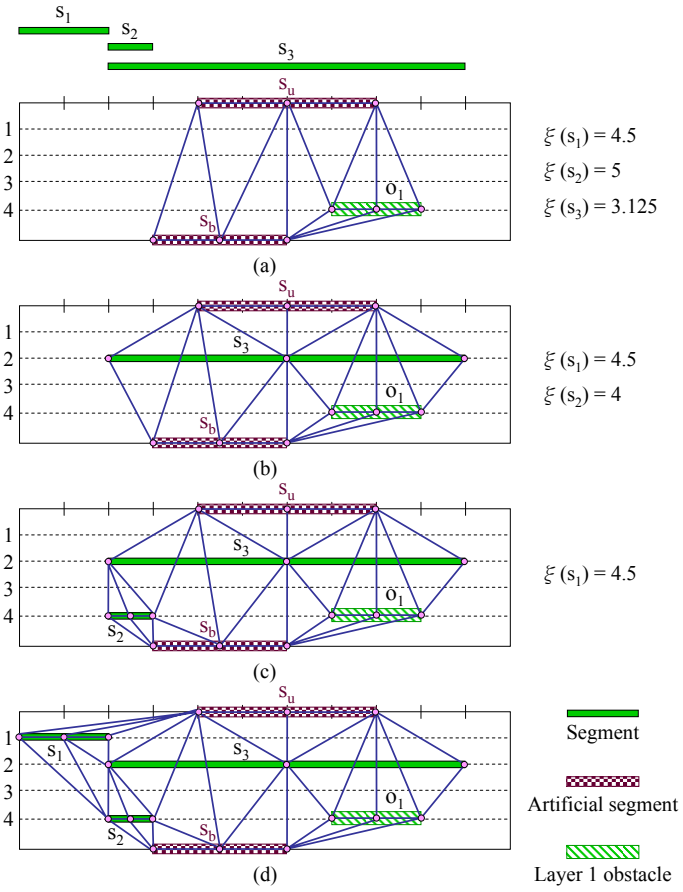


Fig. 8. A density-driven track assignment example. (a) The initial Delaunay triangulation. (b) Track assignment for segment s_3. (c) Track assignment for segment s_2. (d) Track assignment for segment s_1.

IV. Experimental Results

The TTR routing system was implemented in the C++ programming language on a 1.2 GHz SUN Blade-2000 workstation with 8 GB memory. We used the LEDA packages to compute the Voronoi diagrams and Delaunay triangulation. We conducted the experiments based on the 11 MCNC routing benchmarks [3] (these designs have 3–4 routing layers and contain up to 28K connections) and 5 real industrial Faraday benchmarks introduced in [1]. (See Table I for the statistics of the Faraday benchmarks.) In our implementation, the

parameter α in Eq. (1) was set to 0.5, and the parameters β, κ_p, κ_n, B_l, and B_u in Eq. (2) for all benchmarks were given as 0.5, 2, -2, 10%, and 40%, respectively.

TABLE I
THE FARADAY BENCHMARK CIRCUITS.

Circuit	Size (μm^2)	#Layers	#Nets	#Connections	#Pins
DMA	408.4×408.4	6	13256	36162	73982
DSP1	706×706	6	28447	63495	144872
DSP2	642.8×642.8	6	28431	36686	144703
RISC1	1003.6×1003.6	6	34034	95106	196677
RISC2	959.6×959.6	6	34034	95099	196670

We compared the proposed two-pass, top-down routing framework of TTR with the grid-based full-chip multilevel router considering balanced routing density in [20] (named MROR). The MROR program was provided by the authors of [20] and was run on the same machine. For fair comparison, TTR used the same setting for the size of routing tiles in all benchmarks as MROR. Note that as reported in [20], MROR achieves better solutions than the previous work [3], and thus we shall directly compare TTR with MROR.

In addition, we also examined the effects of the Voronoi-diagram-based density critical area analysis (CAA) in TTR by comparing with the minimum-pin density routing algorithm presented in [9]. Note that in [9], the authors applied their algorithm in an ILP-based global router called BoxRouter [8]. Therefore, to focus on the comparison of the two CAA algorithms, we integrated the minimum-pin density routing algorithm into TTR. In other words, we removed the prerouting of TTR and replaced the cost function of the global router in Eq. (2) by the minimum-pin density routing algorithm.

Tables II and III show the comparison results on the MCNC and Faraday benchmarks, respectively. Note that since the MROR program can only handle the designs with all pins lying in layer 1 (as in the MCNC benchmarks), we did not conduct the experiments on the Faraday benchmarks (where pins are distributed between layers 1 and 3) for MROR. In the tables, we used the same metrics as those in [20] which can evaluate the uniformity of wire distribution in the routing stage, where "Rout." stands for routability, "#Net$_{max}$" denotes the maximum number of nets crossing a level-0 tile, "#Net$_{avg_h}$" represents the average number of nets horizontally crossing a tile and "σ_h" gives its standard deviation, and "#Net$_{avg_v}$" gives the average number of nets vertically crossing a tile and "σ_v" gives its standard deviation. For the TTR routing systems, "#LG" denotes the total number of layer groups for the layer assignment, and "#Seg" shows the total number of segments.

As shown in the tables, all routers obtain 100% routing completion on the MCNC benchmarks, and both routers applying the new framework of TTR outperform the multilevel router MROR in wire uniformity. Compared with MROR, TTR incorporated with the minimum-pin density global routing algorithm reduces #Net$_{max}$, #Net$_{avg_v}$, and #Net$_{avg_h}$ by 32%, 28%, 26% respectively, and TTR with Voronoi-diagram-based CAA can achieve 43%, 34%, 36% reductions on #Net$_{max}$, #Net$_{avg_v}$, and #Net$_{avg_h}$ respectively. Moreover, the routers using the TTR framework also result in at least 35% smaller standard deviations of wire distribution in both directions (which implies better density smoothness) than MROR. The results on the Faraday benchmarks also show that the global routing guided by the Voronoi-diagram-based CAA can achieve better wire uniformity than the minimum-pin density global router. Fig. 9 shows the routing layouts of "S13207" and the corresponding wire-crossing maps in the vertical direction for the aforementioned three routers, and Fig. 10 shows the results for the Faraday circuit "RISC1" and the horizontal wire-crossing maps. The experimental results consistently show the superior effectiveness and efficiency of our routing algorithm and framework in wire density control.

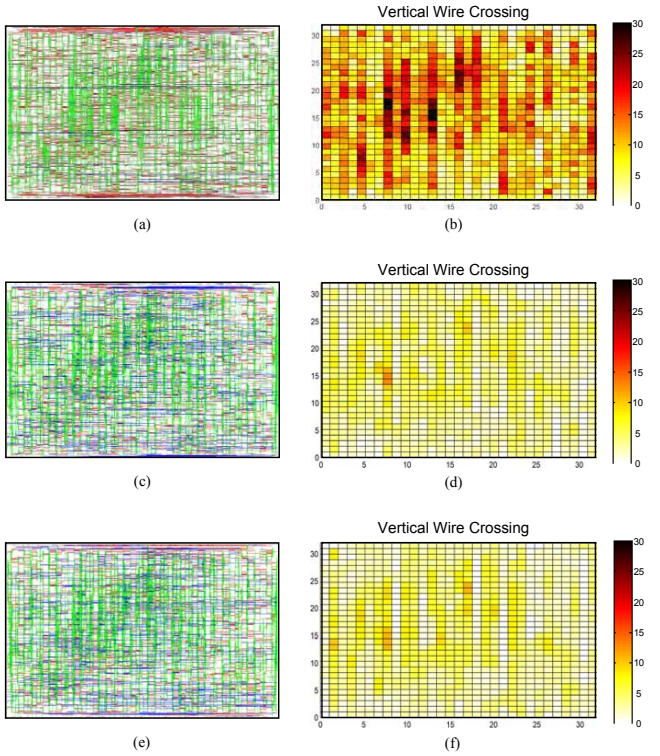


Fig. 9. The routing result and the vertical wire-crossing map in tiles for "S13207." (The red, green, and blue lines represent metals 1, 2, and 3, respectively) (a) and (b) The routing layout and its vertical wire crossing of MROR [20]. The maximum vertical wire crossing is 27. (c) and (d) The routing layout and its vertical wire crossing obtained from the minimum-pin density global routing [9] + TTR's routing framework. The maximum vertical wire crossing is 13. (e) and (f) The routing layout and its vertical wire crossing of TTR (Ours). The maximum vertical wire crossing is only 11.

V. CONCLUSIONS

We have presented a new two-pass, top-down full-chip grid-based router, named TTR, considering wire density for CMP variation control. TTR features a new Voronoi-diagram-based density critical area analyzer, a planarization-aware global router, a layer assigner for panel-density minimization, and an effective track assigner based on the incremental Delaunay triangulation. Experimental results have shown the effectiveness and efficiency of the proposed methods.

REFERENCES

[1] S. N. Adya, S. Chaturvedi, J. A. Roy, D. Papa, and I. L. Markov, "Unification of Partitioning, Floorplanning and Placement," *Proc. ICCAD*, pp. 550–557, Nov. 2004.

[2] S. H. Batterywala, N. Shenoy, W. Nicholls, and H. Zhou, "Track Assignment: A Desirable Intermediate Step Between Global Routing and Detailed Routing," *Proc. ICCAD*, pp. 59–66, Nov. 2002.

[3] Y.-W. Chang and S.-P. Lin, "MR: A New Framework for Multilevel Full-Chip Routing," *IEEE TCAD*, vol. 23, no. 5, pp. 793–800, May 2004.

[4] T.-C. Chen and Y.-W. Chang, "Multilevel Gridless Routing Considering Optical Proximity Correction," *Proc. ASP-DAC*, pp. 1160–1163, Jan. 2005.

[5] T.-C. Chen, Y.-W. Chang, and S.-C. Lin, "A Novel Framework for Multilevel Full-Chip Gridless Routing," *Proc. ASP-DAC*, pp. 636–641, Jan. 2006.

[6] H.-Y. Chen, M.-F. Chiang, Y.-W. Chang, L. Chen, and B. Han, "Novel Full-Chip Gridless Routing Considering Double-Via Insertion," *Proc. DAC*, pp. 755–760, Jul. 2006.

[7] Y. Chen, P. Gupta, and A. B. Kahng, "Performance-Impact Limited Area Fill Synthesis," *Proc. DAC*, pp. 22–27, Jun. 2003.

TABLE II

COMPARISON FOR THE WIRE DENSITY CONTROL ON THE MCNC BENCHMARKS.

Circuit	MROR [20]						Minimum pin density global routing [9] + TTR's routing framework								TTR (Ours)							
	#Net$_{max}$	#Net$_{avg_v}$	#Net$_{avg_h}$	σ_v	σ_h	CPU (sec)	#LG	#Seg	#Net$_{max}$	#Net$_{avg_v}$	#Net$_{avg_h}$	σ_v	σ_h	CPU (sec)	#LG	#Seg	#Net$_{max}$	#Net$_{avg_v}$	#Net$_{avg_h}$	σ_v	σ_h	CPU (sec)
Mcc1	45	9.9	11.3	7.6	7.3	77.4	124	2600	41	10.3	11.1	5.1	7.6	36.1	124	2639	30	10.3	11.0	5.9	6.4	33.4
Mcc2	96	18.7	20.9	17.3	18.5	2714.9	256	15814	119	20.6	22.2	14.4	19.6	798.0	256	16644	87	20.5	22.2	13.9	16.0	645.0
Struct	7	1.4	1.4	1.1	1.6	61.4	193	2128	5	1.2	0.8	0.9	0.8	66.8	167	2124	6	1.1	0.8	1.1	1.0	58.2
Primary1	15	0.7	0.6	1.2	1.8	69.1	328	2423	12	0.8	0.7	0.9	1.4	27.0	215	2207	6	0.7	0.3	0.9	0.8	24.3
Primary2	25	2.1	1.9	1.6	4.5	322.2	387	8338	22	2.5	1.9	1.3	2.8	144.0	303	7693	8	1.8	0.9	1.3	1.6	131.0
S5378	15	4.4	3.5	3.4	2.1	4.5	87	1091	8	2.5	2.4	1.6	1.5	8.1	91	1193	9	2.5	2.4	1.8	1.5	8.2
S9234	14	4.0	2.6	3.2	1.6	3.2	95	912	7	1.7	1.6	1.4	1.3	5.2	95	1003	9	1.7	1.6	1.6	1.2	5.4
S13207	27	9.3	5.9	5.2	2.8	15.8	97	1727	13	3.4	3.0	2.1	1.8	24.8	97	1821	11	3.3	3.0	2.3	1.7	24.2
S15850	26	10.3	7.4	5.4	2.9	23.8	97	1834	12	4.0	3.8	2.3	1.9	34.2	97	1915	13	3.9	3.8	2.4	1.9	33.5
S38417	23	7.3	4.3	4.4	2.2	54.2	188	5043	10	3.0	2.4	1.8	1.4	62.5	188	5462	11	2.9	2.4	2.0	1.4	62.4
S38584	29	9.1	5.8	5.4	2.9	137.7	189	6004	16	3.3	3.1	2.3	1.6	112.0	189	6328	15	3.3	3.1	2.3	1.6	112.0
Comp.	1.00	1.00	1.00	1.00	1.00	1.00	-	-	0.68	0.72	0.74	0.59	0.65	1.01	-	-	0.57	0.66	0.64	0.64	0.65	0.98

TABLE III

COMPARISON FOR THE WIRE DENSITY CONTROL ON THE INDUSTRIAL FARADAY BENCHMARKS.

Circuit	Minimum pin density global routing [9] + TTR's routing framework									TTR (Ours)								
	Rout.	#LG	#Seg	#Net$_{max}$	#Net$_{avg_v}$	#Net$_{avg_h}$	σ_v	σ_h	CPU (sec)	Rout.	#LG	#Seg	#Net$_{max}$	#Net$_{avg_v}$	#Net$_{avg_h}$	σ_v	σ_h	CPU (sec)
DMA	99.19%	272	5168	14	3.14	2.77	1.70	1.77	48.8	99.29%	272	5325	10	3.08	2.70	1.75	1.64	47.0
DSP1	99.11%	264	4241	11	2.91	2.50	1.95	1.89	124.2	99.18%	263	4529	10	2.85	2.44	2.24	1.95	117.3
DSP2	99.10%	268	4676	14	2.78	2.78	1.71	1.92	87.2	99.06%	268	4892	10	2.72	2.70	1.90	1.91	82.3
RISC1	99.16%	265	5864	21	3.63	3.79	2.95	3.78	355.3	99.16%	265	6226	17	3.59	3.73	3.08	3.29	333.4
RISC2	99.23%	260	6141	21	3.64	3.70	2.55	3.08	297.4	99.19%	260	6533	13	3.59	3.62	2.77	2.89	280.0
Comp.	99.16%	-	-	1.00	1.00	1.00	1.00	1.00	1.00	99.18%	-	-	0.75	0.98	0.98	1.08	0.95	0.95

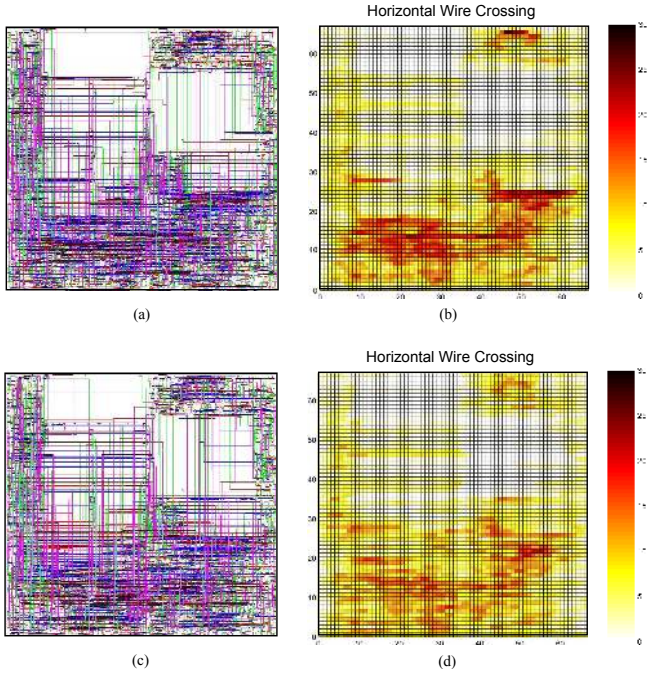


Fig. 10. The routing result and the horizontal wire-crossing map in tiles for "RISC1." (The red, green, blue, magenta, coffee, and aqua blue lines represent metals 1, 2, 3, 4, 5, and 6 respectively, and the white space is allocated by 7 macros.) (a) and (b) The routing layout and its horizontal wire crossing obtained from the minimum-pin density global routing [9] + TTR's routing framework. The maximum horizontal wire crossing is 21. (c) and (d) The routing layout and its horizontal wire crossing of TTR (Ours). The maximum horizontal wire crossing is only 17.

[8] M. Cho, and D. Z. Pan, "A New Global Router Based on Box Expansion and Progressive ILP," *Proc. DAC*, pp. 373–378, Jul. 2006.

[9] M. Cho, D. Z. Pan, H. Xiang, and R. Puri, "Wire Density Driven Global Routing for CMP Variation and Timing," *Proc. ICCAD*, pp. 487–492, Nov. 2006.

[10] J. D. Cho, S. Raje, and M. Sarrafzadeh, "Approximation for the Maximum Cut, k-Coloring and Maximum Linear Arrangement Problems," *Manuscript*, Dept. of EECS, Northwestern Univ., 1993.

[11] J. Cong, J. Fang, and K. Y. Khoo, "DUNE–A Multilayer Gridless Routing System," *IEEE TCAD*, vol. 20, no. 5, pp. 633–647, May 2001.

[12] J. Cong, M. Xie, and Y. Zhang, "An Enhanced Multilevel Routing System," *Proc. ICCAD*, pp. 51–58, Nov. 2002.

[13] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer, 1997.

[14] T.-Y. Ho, Y.-W. Chang, S.-J. Chen, and D.-T. Lee, "Crosstalk- and Performance-Driven Multilevel Full-Chip Routing," *IEEE TCAD*, vol. 24, no. 6, pp. 869–878, Jun. 2005.

[15] T.-Y. Ho, C.-F. Chang, Y.-W. Chang, and S.-J. Chen, "Multilevel Full-Chip Routing for the X-Based Architecture," *Proc. DAC*, pp. 597–602, Jun. 2005.

[16] A. B. Kahng, G. Robins, A. Singh, and A. Zelikovsky, "Filling Algorithms and Analyses for Layout Density Control," *IEEE TCAD*, vol. 18, no. 4, pp. 445–462, Apr. 1999.

[17] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "Pattern Routing: Use and Theory for Increasing Predictability and Avoiding Coupling," *IEEE TCAD*, pp. 777–790, Nov. 2002.

[18] A. Kurokawa, T. Kanamoto, T. Ibe, A. Kasebe, W. F. Chang, T. Kage, Y. Inoue, and H. Masuda, "Dummy Filling Methods for Reducing Interconnect Capacitance and Number of Fills," *Proc. ISQED*, pp. 586–591, Mar. 2005.

[19] K.-S. Leung, "SPIDER: Simultaneous Post-Layout IR-Drop and Metal Density Enhancement with Redundant Fill," *Proc. ICCAD*, pp. 33–38, Nov. 2005.

[20] K. S.-M. Li, Y.-W. Chang, C.-L. Lee, C. Su, and J. E. Chen, "Multilevel Full-Chip Routing with Testability and Yield Enhancement," *IEEE TCAD*, Sep. 2007.

[21] E. Papadopoulou and D. T. Lee, "Critical Area Computation via Voronoi Diagrams," *IEEE TCAD*, vol. 18, no. 4, pp. 463–474, Apr. 1999.

[22] T. H. Park, "Characterization and Modeling of Pattern Dependencies in Copper Interconnects for Integrated Circuits," *Ph.D. Dissertation*, Dept. of EECS, MIT, May 2002.

[23] R. Tian, D. F. Wong, and R. Boone, "Model-Based Dummy Feature Placement for Oxide Chemical-Mechanical Polishing Manufacturability," *Proc. DAC*, pp. 667–670, Jun. 2000.

[24] Taiwan Semiconductor Manufacturing Company (TSMC), *Reference Flows 7.0*.

[25] X. Wang, C. C. Chiang, J. Kawa, and Q. Su, "A Min-Variance Iterative Method for Fast Smart Dummy Feature Density Assignment in Chemical-Mechanical Polishing," *Proc. ISQED*, pp. 258–263, Mar. 2005.

[26] D. White and B. Moore, "An 'Intelligent' Approach to Dummy Fill," *EE Times*, Jan. 3, 2005.