Multilevel Full-Chip Gridless Routing Considering Optical Proximity Correction *

Tai-Chen Chen[1] *and Yao-Wen Chang*[2]

[1]Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan

[2]Department of Electrical Engineering & Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan

tcchen@eda.ee.ntu.edu.tw, ywchang@cc.ee.ntu.edu.tw

Abstract

To handle modern routing with nanometer effects, we need to consider designs of variable wire widths and spacings, for which gridless routers are desirable due to their great flexibility. The gridless routing is much more difficult than the grid-based one because the solution space of gridless routing is significantly larger than that of grid-based one. In this paper, we present *the first* multilevel, full-chip gridless *detailed* router. The router integrates global routing, detailed routing, and congestion estimation together at each level of the multilevel routing. It can handle non-uniform wire widths and consider routability and optical proximity correction (OPC). Experimental results show that our approach obtains significantly better routing solutions than previous works. For example, for a set of 11 commonly used benchmark circuits, our approach achieves 100% routing completion for all circuits while the famous state-of-the-art three-level routing and multilevel routing (multilevel global routing + flat detailed routing) cannot complete routing for any of the circuits. Besides, experimental results show that our multilevel gridless router can handle non-uniform wire widths efficiently and effectively (still maintain 100% routing completion for all circuits). In particular, our OPC-aware multilevel gridless router archives an average reduction of 11.3% pattern features and still maintains 100% routability for the 11 benchmark circuits.

1 Introduction

Research in VLSI routing has received much attention in the literature. Routing is typically a very complex combinatorial problem. In order to make it manageable, the routing problem is usually solved using the two-stage approach of global routing followed by detailed routing. Many routing algorithms adopt a flat framework of finding paths for all nets [1, 16, 22, 29, 31]. The major problem of the flat frameworks lies in their scalability for handling larger designs. As technology advances, technology nodes are getting smaller, and circuit sizes are getting larger. To cope with the increasing complexity, researchers proposed to use hierarchical approaches to handle the problem [4, 15, 28, 32]. The two-level, hierarchical routing framework, however, is still limited in handling the dramatically growing complexity in current and future IC designs which may contain hundreds of millions of gates in a single chip. As pointed out in [10], for a 0.07 μm process technology, a 2.5 $\times$ 2.5 cm^2 chip may contain over 360,000 horizontal and vertical routing tracks. To handle such high design complexity, the two-level, hierarchical approach becomes insufficient. Therefore, it is desired to employ more levels of routing for larger IC designs.

1.1 Multilevel Routing

The multilevel framework has attracted much attention in the literature recently. It employs a two-stage technique: coarsening followed by uncoarsening. The coarsening stage iteratively groups a set of circuit components (e.g., circuit nodes, cells, modules, routing tiles, etc) based on a predefined cost metric until the number of components being considered is smaller than a threshold. Then, the uncoarsening stage iteratively ungroups a set of previously clustered circuit components and refines the solution by using a combinatorial optimization technique (e.g., simulated annealing, local refinement, etc). The multilevel framework has been successfully applied to VLSI physical design. For example, the famous multilevel partitioners, *ML* [2], *hMETIS* [20], and *HPM* [9], the multilevel placer, *mPL* [3], and the multilevel floorplanner/placer, *MB*-tree* [23], all show the promise of the multilevel framework for large-scale circuit partitioning, placement, and floorplanning.

A framework similar to multilevel routing was presented in [14, 25, 27]. Lin, Hsu, and Tsai in [25] and Hayashi and Tsukiyama in [14] presented hybrid hierarchical *global* routers for multi-layer VLSI's [14], in which both bottom-up (coarsening) and top-down (uncoarsening) techniques were used for global routing. Marek-Sadowska [27] proposed a global router based on outer-most loop approach. The approach is similar to the coarsening stage of multilevel routing. Recently, Cong et al. proposed a pioneering multilevel global-routing approach for large-scale, full-chip, routability-driven routing [10]. Cong et al. later proposed an enhanced multilevel routing system, named MARS [11], which incorporates resource reservation, a graph-based Steiner tree heuristic, and a history-based multi-iteration scheme to improve the quality of the multilevel routing algorithm in [10]. The final results of both of the multilevel global routing are tile-to-tile paths for all the nets. The results are then fed into a non-multilevel gridless detailed router, called DUNE [8], to find the exact connection for each net. (Therefore, MARS is in fact a multilevel *global* router, but not a *detailed* router.) Lin and Chang also proposed a multilevel approach for full-chip, *grid-based* routing, which considers both routability and performance [5, 26]. This framework integrates grid-based global routing, detailed routing, and resource estimation together at each level, leading to more accurate routing resource estimation during coarsening and thus facilitating the solution refinement during uncoarsening. Their experimental results show the best routability among the previous works for grid-based routing. Recently, Ho et al. in [17] and [18] presented another multilevel framework for full-chip, grid-based routing considering crosstalk and antenna effects. The framework incorporates an intermediate stage of layer/track assignment between the coarsening stage and the uncoarsening stage. The coarsening stage performs only global routing while global routing as well as detailed routing are integrated together at the uncoarsening stage.

Most of the previous routing algorithms are grid-based, assuming uniform grids and wire sizes. However, the grid-based approach is not effective to handle modern routing problems with nanometer effects, such as OPC. To cope with these nanometer effects, we need to consider designs of variable wire widths and spacings, for which gridless routers are desirable due to their great flexibility (although they are limited by their high complexity). The gridless routing is much more difficult than the grid-based routing because the solution space of gridless routing is significantly larger than that of grid-based routing.

1.2 OPC Technologies

Microlithography has become one of the key techniques for the nanometer process technology. The sub-wavelength lithography introduces a huge burden in the manufacturing process because the diffraction of light physically limits the critical dimension (CD), such as the shortest length of the gate channel. The distortions in optical lithography include corner rounding, pulling back at the end of the narrow line, and the wide variation of line widths. To compensate for the distortion or cancel out the interference from the neighboring light diffractions, two techniques–OPC [12, 13, 33] and Phase Shift Masking (PSM) [6, 7, 24, 30] have been demonstrated to show significant improvements in sub-wavelength lithography technology. OPC works by adding or subtracting some features as serifs or line segments while PSM operates by changing the phase of the transmitted light through certain regions on the mask. However, a rule-based OPC system that decides the added OPC features based on geometrical rules cannot handle rules that are too complicated; a model-based OPC system that requires many iterations of simulations takes a long time to choose the sizes and positions of the added features; and PSM increases the cost of the masks by introducing the phase shift mask.

Under sub-wavelength lithography, it is very hard to obtain exactly the right image we desire on the wafer. That is, due to the diffraction of exposure light, there may be a great number of variations of the final image on resist compared to the designed image on the mask. Using some optics simulation software, we can clearly see this variation as shown in Figure 1(a). These variations can be classified into mainly three types: line (a line is a horizontal/vertical segment of a net or is a via) width shrinking, line end shortening, and corner rounding (as shown in Figure 1(b)∼(d)). For each type of variation, we can add pattern features to compensate for the distortions and acquire what we really need on the resist. We can add *Line Biasings* in line sides to compensate for its shrinking, add *Hammerheads* at line ends to compensate for its shortening, and add *Serifs* at corners to make the angle more right. More pattern features were added, implying that we need more data volume to record these features. If we can avoid more variations in the routing stage, we can reduce more data volume when we do OPC. Therefore, it is desirable to incorporate the OPC consideration into router designs.

As a relatively new issue, there is not much work in the literature on routing with the OPC consideration. Recently, Huang and Wong presented a pioneering work on OPC-friendly maze routing based on the Lagrangian relation formulation [19]. The router is grid-based and considers only two-pin connections. Further, it uses the flat framework and thus handles only hundreds of connections.

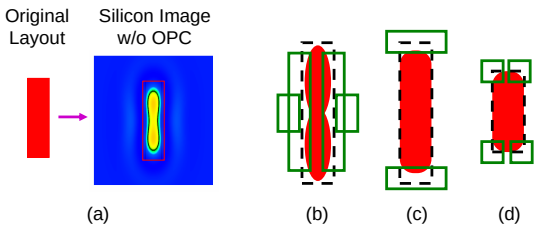


Figure 1: (a) Optical proximity effects. Three major OPC techniques: (b) Line Biasing; (c) Hammerhead; (d) Serif.

1.3 Our Contribution

In this paper, we propose the *first* multilevel full-chip, *gridless* detailed router. The three main features of the proposed method are as follows: (1) the first multilevel gridless router that integrates gridless global and detailed routing; (2) a multilevel gridless router that can handle non-uniform wire widths; (3) an OPC-aware, routability-driven multilevel gridless router that can optimize routing completion rate and reduce OPC pattern feature requirements.

*This work was partially supported by the SpringSoft, Inc. and the National Science Council of Taiwan under Grant No. NSC 93-2215-E-002-009 & NSC 93-2215-E-002-029.

Experimental results show that our approach obtains significantly better routing solutions than the three-level routing [8] and the multilevel routing (multilevel global routing + flat detailed routing) [10, 11]. For the 11 benchmark circuits provided by the authors of [10], our approach obtains 100% routing completion for all circuits while the three-level routing and the multilevel routing cannot complete routing for any of the 11 circuits. Besides, experimental results show that our multilevel gridless router can handle non-uniform wire widths efficiently and effectively (still maintain 100% routing completion for all circuits). In particular, our OPC-aware multilevel gridless router archives an average reduction of 11.3% pattern features and still maintains 100% routability for the 11 benchmark circuits.

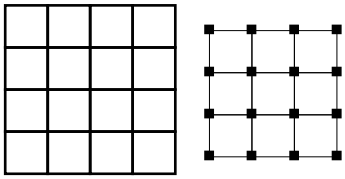
The rest of this paper is organized as follows. Section 2 presents the routing model and the multilevel routing framework. Section 3 presents our framework for routability-driven and OPC-aware routing. Experimental results are shown in Section 4. Finally, we give concluding remarks in Section 5.

2 Preliminaries

2.1 Routing Model

Routing in modern IC's is a very complex process. Thus, we can hardly obtain desired solutions directly. Our routing algorithm is based on a graph-search technique guided by the congestion information associated with routing regions and topologies. The router assigns higher costs to route nets through congested areas to balance the net distribution among routing regions.

Before we can apply the graph-search technique to multilevel routing, we first need to model the routing resource as a graph such that the graph topology can represent the chip structure. Figure 2 illustrates the graph modelling. For the



(a) (b)

Figure 2: Routing graph: (a) partitioned layout and (b) routing graph.

modelling, we first partition a chip into tiles. A node in the graph represents a tile in the chip, and an edge denotes the boundary between two adjacent tiles. Each edge is assigned a capacity according to the width/height of a tile. The graph is used to represent the routing structure and is called multilevel routing graph G_0. A global router finds tile-to-tile paths for all nets on G_0 to guide the detailed router. The goal of global routing is to route as many nets as possible while meeting the capacity constraint of each edge and any other constraint, if specified. Note that, because of the gridless nature of our routing problem, the cost of routing a net is associated with the wire width and spacing. As the process technology advances, multiple routing layers are possible. Wires in each layer run either horizontally or vertically. We refer to the layer as a horizontal (H) or a vertical (V) routing layer.

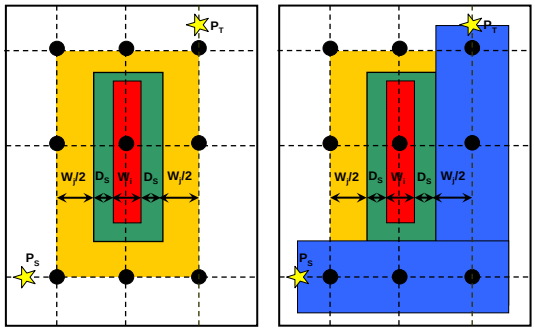
In the detailed routing stage, seeking a design-rule-correct path between two given points in the routing region is our major concern. At first, for each obstacle (a pre-routed wire or a pin), we extend a range which is the sum of the obstacle (wire/via) spacing and a half of the width of the routing wire. As shown in Figure 3(a), the extended range is the sum of D_S and $W_j/2$, where D_S and W_j are the design rules of wire/via spacing and the width of the routing wire, respectively. According to the boundaries of all extended regions and the center of the obstacle, we get three x-coordinates and three y-coordinates for each extended region. We store all x-coordinates and y-coordinates of all extended regions and all centers of all obstacles into two sorted array, CG_x and CG_y, separately. Based on CG_x and CG_y, we can construct a *connection graph*, (CG). A node in a CG denotes that it is an intersection of one x-coordinate in CG_x and one y-coordinate in CG_y. A node in a CG also denotes that it is a feasible point for routing the wire. As shown in Figure 3(a), the black circles in the intersection are the feasible points for routing the wire . Therefore, to seek for a design-rule-correct path from the source, P_S, to the target, P_T, we just need to search all feasible points and find if a successful path exists. As shown in Figure 3(b), a design-rule-correct path from P_S to P_T is found through the five feasible points shown on the P_S-P_T path.

2.2 Multilevel Routing Model

Figure 4 shows our multilevel framework. As illustrated in Figure 4, G_0 corresponds to the routing graph of the level 0 of the multilevel coarsening stage. At each level, our global router first finds routing paths for the *local nets* (or *local 2-pin connections*) (those nets [connections] that entirely sit inside a tile), and then the detailed router is used to determine the exact wiring. After the global and detailed routing are performed, we merge four adjacent tiles of G_0 into a larger tile and at the same time perform congestion estimation for use at the next level (i.e., level 1 here). Coarsening continues until the number of tiles at a level, say the k-th level, is below a threshold. After finishing coarsening, the uncoarsening stage tries to refine the routing solution starting from the last level k where coarsening stops. During uncoarsening, the unroutable nets during coarsening are considered, and maze routing and rip-up and re-route are performed to refine the routing solution. Then we proceed to the next level (level $k-1$) of uncoarsening by expanding each tiles to four finer tiles. The process continues up to level 0 when the final routing solution is obtained.

3 Multilevel Routing Framework

Our multilevel routing framework is inspired by the work [5, 26]. Nevertheless, our routing model is totally different from [5, 26]. The works [5, 26] are grid-based routing, and thus they can apply a graph-searching technique (e.g., breadth-first search or depth-first search) on pre-defined grids. However, our



(a) (b)

Figure 3: (a) We build the CG and locate the feasible points by extending a range which is the sum of the obstacle (wire/via) spacing and a half of the width of the routing wire. D_S, W_i, and W_j are the design rule of wire/via spacing, the width of the pre-routed wire, and the width of the routing wire, respectively; (b) A design-rule-correct path from the source, P_S, to the target, P_T, is found through the five feasible points shown on the P_S-P_T path.

work is based on gridless routing. Since the solution space of gridless routing is significantly larger than that of grid-based routing, we apply a different routing model for gridless routing, as discussed in Section 2.1. Besides, our congestion estimation is significantly different from [5, 26].

Our router tends to route wider nets first since a wider net enjoys more routing resource. Beside, our router tends to route shorter nets first since we route local nets at each level of coarsening. It is obvious that the local nets at the lower level (say, level 0) are usually shorter than those at a higher level (say, level k). Naturally, a shorter net enjoys less freedom while searching for a path to route it. This fact holds even during rip-up and re-route. Thus, this observation implicitly suggests that a shorter net has a higher priority than a longer net as far as routability is concerned. Kastner, Bozorgzadeh, and Sarrafzadeh in [21] also suggest this conclusion. Thought this net ordering scheme may not be the optimal solution for some routing problems (for example, when timing is considered, routing the most critical net first often leads to better timing performance), it is still a reasonable alternative.

3.1 Routability-Driven Multilevel Routing

Given a netlist, we first run the minimum spanning tree (MST) algorithm to construct the topology for each net, and then decompose each net into 2-pin connections, with each connection corresponding to an edge of the minimum spanning tree. Our multilevel framework starts from coarsening the finest tiles of level 0. At each level, tiles are processed one by one, and only local nets (connections) are routed. At each level, the two-stage routing approach of global routing followed by detailed routing is applied.

The global routing is based on the approach used in the Pattern Router [21] and first routes local nets (connections) on the tiles of level 0. Let the multilevel routing graph of level i be $G_i = (V_i, E_i)$. Let $R_e = \{\ e \in E_i \mid e$ is the edge chosen for routing$\}$. We apply the cost function $\alpha : E_i \rightarrow \Re$ to guide the routing:

$$\alpha(R_e) = \max_{e \in R_e} c_e, \tag{1}$$

where c_e is the congestion of edge e and is defined by

$$c_e = \frac{p_e}{d_e},$$

where p_e and d_e are the capacity and density associated with e, respectively. Pattern routing uses an L- or Z-shaped route to make the connection, which gives the shortest path length between two points. Therefore, the wire length is minimum. We measure the routing congestion based on the *channel density* defined by the sum of wire spacing and wire width for gridless routing. (Note that the definition is different from the case in grid-based routing, for which channel density is defined as the maximum number of parallel nets passing through a routing channel.) If pattern routing fails, we give up routing the connection. We refer to a *failed net* (*failed connection*) as that causes an overflow. The failed nets (connections) will be reconsidered (refined) at the uncoarsening stage.

After the global routing is completed, we perform detailed routing with the guidance of the global-routing results and find a real path in the chip. Our detailed router is based on Dijkstra's shortest path algorithm and supports the *local refinement*. After the detailed routing finishes routing a net, the channel density associated with an edge of a multilevel graph is updated accordingly. This is called *congestion estimation*. There are at least two advantages by using this approach. First, routing congestion estimation is more accurate than that performing global routing alone since we can precisely evaluate the routing region. Second, we can obtain a good initial solution for the following refinement very effectively since pattern routing enjoys very low time complexity and uses fewer routing resources due to its simple L- and Z-shaped routing patterns.

The uncoarsening stage starts to refine each local failed net (connection), left from the coarsening stage. The global router is now changed to the maze router with the same cost function in the coarsening stage. Uncoarsening continues until the first level G_0 is reached and the final solution is found. Note that the global maze routing here serves as an elaborate rip-up and re-route processor, in contrast to the simple L- and Z-shaped routing during coarsening. (For rip-up and re-route in our multilevel routing algorithm, we mean the maze routing at the uncoarsening stage. It is only applied to global routing for better efficiency and quality trade-off.) This two-stage approach of global and local refinement of detailed routing gives our overall refinement scheme.

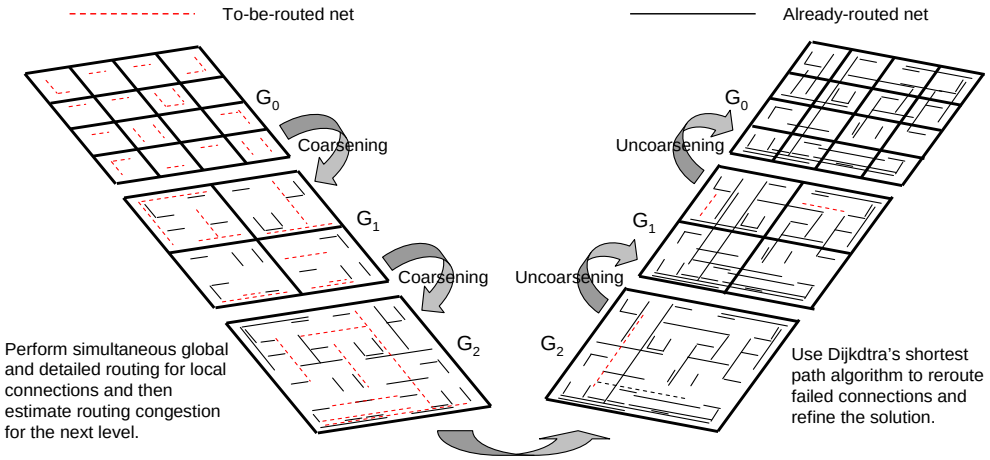


Figure 4: The multilevel framework flow.

3.2 OPC-Aware Multilevel Routing

In modern nanometer process technologies, such as 90 nm technology and beyond, most of the metal layers need OPC to control the line width and length variations. Considering OPC in the routing stage, we can maximize the effects of the correction and thus reduce the number of OPC pattern features during masking.

3.2.1 OPC Cost Calculation

As shown in Figure 1(d), for a line, we need to add four serifs at the corners to increase the fidelity of images. As the length of a line increases, the ends of the line are shortened. Therefore, as shown in Figure 1(c), we need to add two hammerheads at the line ends for a *long* line. (We say a line to be a long line if its length is longer than or equal to L_T, where L_T is the threshold length for a long line and is defined by the foundry.) Besides, the overlapping length of a line with neighboring lines may increase as the length of the line increases; further, a wider line is easier to be affected by neighboring lines than a narrower one. These phenomena make the sides of a line shrink more seriously. Therefore, as shown in Figure 1(b), we need some line biasings in the line sides to correct the optical proximity effects for a line. The total number of line biasings for a line is determined by the length and width of the line. According to the above modelling, we define the OPC cost of a line e whose length and width are l_e and w_e, respectively, to be the total number of pattern features as follows:

$$cost(e) = \begin{cases} 4 + f(l_e, w_e) & \text{when } l_e < L_T \\ 6 + f(l_e, w_e) & \text{otherwise,} \end{cases}$$

where f is a step function and is defined as follows:

$$f(l_e, w_e) = 2 \times (\lfloor l_e/L_L \rfloor + \lfloor w_e/W_L \rfloor). \quad (2)$$

Here, L_L and W_L (the unit-length and unit-width for adding a pair of line biasings) are parameters related to the process technology and are defined by the foundry. Therefore, the total OPC cost for a *connection* is the sum of the OPC costs for lines that belong to the connection.

3.2.2 OPC Cost Minimization

Based on Dijkstra's shortest path algorithm (as shown in Figure 5(a)), we apply the following algorithm, called *SPOM* (*Simultaneous Pathlength and OPC cost Minimization*), to find a shortest path with the minimum number of pattern features. It associates each basic detailed routing region u with two labels: $dist(u)$ and $feature(u)$, where $dist(u)$ is the distance of the shortest path from source s to u, and $feature(u)$ is the minimum number of pattern features along the shortest path from s to u. Initialize $dist(u) = \infty, feature(u) = \infty, \forall u \neq s, dist(s) = 0$, and $feature(s) = 0$. The computation of label $dist$s is the same as the original Dijkstra's algorithm. The computation of $feature(v)$ is shown in Figure 5(b), where $w(u, v)$ and $o(u, v)$ are the distance and the number of additional pattern features between nodes u and v, respectively.

```
Dijkstra(G, w, s)
1 Initialize-Single-Source(G, s);
2 S ← ∅;
3 Q ← V[G];
4 while Q ≠ ∅
5     u ← Extract-Min(Q);
6     S ← S ∪ {u};
7     for each vertex v ∈ Adj[u]
8         Relax(u, v, w);
```

(a)

```
RELAX(u, v, w)
1 if dist(v) ≥ dist(u) + w(u, v)
2     dist(v) ← dist(u) + w(u, v);
3     if feature(v) ≥ feature(u) + o(u, v)
4         feature(v) ← feature(u) + o(u, v);
5         Record u as the predecessor routing region of v.
```

(b)

Figure 5: (a) Dijkstra's shortest path algorithm. (b) The algorithm to compute $feature(v)$.

The basic idea is to compare the distance label $dist$s first and then compare the pattern feature number label $feature$s. The value $feature(v)$ of a neighboring routing region v with $dist(v) < dist(u)$ stays unchanged because the path from s through u to v is not the shortest path between s and v. Note that it is possible that there may exist several shortest paths with different numbers of pattern features. It is clear that SPOM algorithm guarantees finding a shortest path with the minimum number of pattern features, if such a path exists.

Circuit	Size (μm)	#Layers	#Nets	#Pins
Mcc1	45000×39000	4	1693	3101
Mcc2	161482×161482	4	7541	25024
Struct	4903×4904	3	3551	5471
Prim1	7522×4988	3	2037	2941
Prim2	10438×6488	3	8197	11226
S5378	435×239	3	3124	4818
S9234	404×225	3	2774	4260
S13207	660×365	3	6995	10776
S15850	705×389	3	8321	12793
S38417	1144×619	3	21035	32344
S38584	1295×672	3	28177	42931

Table 1: The benchmark circuits.

4 Experimental Results

We have implemented our multilevel gridless routing system in the C++ language on a 900 MHz SUN Blade-2000 workstation with 8 GB memory. We compared our results with the gridless routers presented in [8] and [10] based on the 11 benchmark circuits provided by the authors. (Note that the recent multilevel routers presented in [5, 17, 18, 26] are all grid-based.) The design rules for wire/via widths and wire/via separation for detailed routing are the same as those used in [8, 10].

Table 1 lists the set of benchmark circuits. In the table, "Circuit" gives the names of the circuits, "Size" gives the layout dimensions, "#Layers" denotes the number of routing layers used, and "#Nets" gives the number of two-pin connections after net decomposition.

4.1 Routability-Driven Multilevel Routing with Uniform Nets

Table 2 gives the comparison of our multilevel gridless routing for routability with the three-level routing [8] and the multilevel routing (multilevel global routing + classical detailed routing) [10]. The three-level routing (A) first uses a performance-driven global router, then a noise-constrained wire spacing and track assignment algorithm, and a detailed router [8]. The multilevel routing (B) gives the main results from [10]. In the table, "Time (s)" represents the running times in second, "#Rtd. Nets" denotes the number of routed nets, "Comp. Rates" gives the routing completion rates, and "avg." (bottom) denotes the average routing completion rates. As shown in the table, our multilevel gridless routing obtains significantly better routing solutions than the three-level routing [8] and the multilevel routings [11]. For the 11 benchmark circuits provided by the authors of [11], our multilevel gridless routing obtains 100% routing completion for all circuits while the three-level routing and the multilevel routing cannot complete routing for any of the 11 circuits. The reason that we can achieve 100% routing completion for all circuits while [11] cannot do it is that our framework integrates grid-based global routing, detailed routing, and congestion estimation together at each level, leading to more accurate routing congestion estimation during coarsening and thus facilitating the solution refinement during uncoarsening. Because the problem of gridless routing is much difficult than that of grid-based routing, it is not fair to compare a girdless routing system with a grid-based routing system. Therefore, we do not compare our work with [5, 26]. (Despite [5, 26] can achieve 100% routing completion for all circuits, they are grid-based routing systems.)

4.2 Routability-Driven Multilevel Routing with Non-uniform Nets

We also performed experiments on the benchmark circuits with non-uniform wire widths. Since no previous works on multilevel routing with non-uniform wire widths are available for comparative studies, we generate the benchmarks with non-uniform wire widths by modifying the original benchmark circuits with uniform wire widths. (Note that because the benchmark circuits S5378∼S38584 are standard-cell designs, widening any pin is violating the design rules for via spacing. Therefore, we do not run these benchmark circuits in this experiment.) We widened the longest 20% of nets to 120% of the original wire width. As shown in Table 3, our multilevel gridless router still achieves 100% routing completion for all circuits, with some overheads in the running time.

Circuit	(A) Three-Level Routing [8]			(B) Multilevel Routing of [11]			(C) Our Results		
	Time (s)	#Rtd. Nets	Cmp. Rates	Time (s)	#Rtd. Nets	Cmp. Rates	Time (s)	#Rtd. Nets	Cmp. Rates
Mcc1	933.2	1499	88.0%	182.5	1677	99.0%	190.2	1693	100%
Mcc2	12333.6	5451	72.3%	4367.4	7495	99.9%	3711.0	7541	100%
Struct	406.2	3530	99.4%	529	3549	99.9%	6.5	3551	100%
Prim1	239.1	2018	99.0%	22.6	2036	99.9%	5.1	2037	100%
Prim2	1331	8109	98.9%	173.5	8195	99.9%	46.7	8197	100%
S5378	430.2	2607	83.4%	34.4	3116	99.7%	45.6	3124	100%
S9234	355.2	2467	88.9%	24.4	2771	99.9%	25.1	2774	100%
S13207	1099.5	6118	87.5%	115.4	6983	99.8%	136.2	6995	100%
S15850	1469.1	7343	88.2%	154.6	8311	99.9%	362.2	8321	100%
S38417	3560.9	19090	90.8%	567.6	20993	99.8%	403.1	21035	100%
S38584	7086.5	25642	91.0%	1308.2	28131	99.8%	765.1	28177	100%
avg.			89.8%			99.7%			100%

Table 2: Comparison among (A) the three-level routing [8], (B) the multilevel routing [11], and (C) our multilevel gridless routing. Note: (A) and (B) were run on a 440 MHz Sun Ultra-5 with 384 MB memory; (C) was run on a 900 MHz Sun Blade-2000 with 8 GB memory.

Circuit	Without OPC				With OPC				
	Time (s)	#Pattern Features	#Rtd. Nets	Cmp. Rates	Time (s)	#Pattern Features	#Rtd. Nets	Cmp. Rates	Impr. on #Pattern Features
Mcc1	190.2	40656	1693	100%	207.5	38102	1693	100%	6.7%
Mcc2	3711.0	208132	7541	100%	4330.3	201901	7541	100%	3.1%
Struct	6.5	149142	3551	100%	7.1	130810	3551	100%	14.0%
Prim1	5.1	41110	2037	100%	5.8	37012	2037	100%	11.1%
Prim2	46.7	98364	8197	100%	48.8	88214	8197	100%	11.5%
S5378	45.6	78104	3124	100%	47.8	73092	3124	100%	6.9%
S9234	25.1	55982	2774	100%	26.2	50122	2774	100%	11.7%
S13207	136.2	154098	6995	100%	158.0	130230	6995	100%	18.4%
S15850	362.2	199704	8321	100%	392.9	170238	8321	100%	17.4%
S38417	403.1	631032	21035	100%	495.0	580258	21035	100%	8.8%
S38584	765.1	676250	28177	100%	783.0	591232	28177	100%	14.4%
avg.				100%				100%	11.3%

Table 4: Results of our OPC-aware multilevel gridless routing.

Circuit	Time (s)	#Rtd. Nets	Cmp. Rates
Mcc1	257.0	1693	100%
Mcc2	5946.2	7541	100%
Struct	9.9	3551	100%
Prim1	10.4	2037	100%
Prim2	98.8	8197	100%
avg.			100%

Table 3: Results of our multilevel gridless routing for non-uniform nets. Note: Because the benchmark circuits S5378∼S38584 violate the design rules for via spacing, we do not list these cases in the table.

4.3 OPC-Aware Multilevel Routing

Finally, we performed experiments on OPC-aware routing. For OPC-aware routing, no previous routers are available to us for comparative studies (since this is a relatively new topic for research). (The only work on OPC-aware routing is by Huang and Wong [19], which is a flat, grid-based maze router that handles only hundreds of two-pin connections.) We refer to a line as a *long* line if its length is 5 times larger than the minimum wire width. Besides, we set the L_L/W_L to be 5/1 times of the minimum wire width. The results are listed in Table 4. In the table, "Time (s)" represents the running times in second, "#Pattern Features" denotes the total number of pattern features, "#Rtd. Nets" denotes the number of routed nets, "Comp. Rates" gives the routing completion rates, "Impr. on #Pattern Features" denotes the improvement of the total number of pattern features, and "avg." (bottom) denotes the average routing completion rates. Compared with our routibility-driven multilevel gridless router, the experimental results show that our OPC-aware multilevel gridless router achieves an average 11.3% reduction in the number of pattern features required and still maintains 100% routing completion for all circuits, with very small overheads in the running time. The results show the effectiveness of our OPC-aware multilevel router.

5 Conclusion

In this paper, we have proposed the first multilevel, full-chip gridless detailed router. The router can handle non-uniform wire widths and consider routability and OPC. Experimental results have shown that our approach obtains significantly better routing solutions than previous works. Besides, experimental results have also shown that our multilevel gridless router is very effective in handling non-uniform wire widths and the OPC effect.

References

[1] C. Albrecht, "Global routing by new approximation algorithms for multicommodity flow," *IEEE Trans. CAD*, vol. 20, no. 5, pp. 622–632, May 2001.

[2] C. J. Alpert, J.-H. Huang, and A. B. Kahng, "Multilevel circuit partitioning," *IEEE Trans. CAD*, vol. 17, no. 8, pp. 655–667, August 1998.

[3] T. Chan, J. Cong, T. Kong, and J. Shinnerl, "Multilevel optimization for large-scale circuit placement," *Proc. ICCAD*, pp. 171–176, Nov. 2000.

[4] Y.-W. Chang, K. Zhu, and D.-F. Wong, "Timing-driven routing for symmetrical-array-based FPGAs," *ACM Trans. Design Automation of Electronic Systems*, vol. 5, no. 3, pp. 433–450, July 2000.

[5] Y.-W. Chang and S.-P. Lin, "MR: A new framework for multilevel full-chip routing," *IEEE Trans. CAD*, vol. 23, no. 5, pp. 793–800, May 2004.

[6] N. Cobb and A. Zakhor, "Large Area Phase-Shift mask Design," *SPIE*, 2197:348–359, 1994.

[7] N. Cobb, A. Zakhor, and Eugene Miloslavsky, "Mathematical and CAD Framework for Proximity Correction," *SPIE*, 2726:208–222, 1996.

[8] J. Cong, J. Fang, and K. Khoo, "DUNE: A multi-layer gridless routing system with wire planning," *Proc. ISPD*, pp. 12–18, April 2000.

[9] J. Cong, S. Lim, and C. Wu, "Performance driven multilevel and multiway partitioning with retiming," *Proc. DAC*, pp. 274–279, June 2000.

[10] J. Cong, J. Fang, and Y. Zhang, "Multilevel approach to full-chip gridless routing," *Proc. ICCAD*, pp. 396–403, Nov. 2001.

[11] J. Cong, M. Xie, and Y. Zhang, "An enhanced multilevel routing system," *Proc. ICCAD*, pp. 51–58, Nov. 2002.

[12] C.-C. Fu, T.-S. yang, and Douglas R. Stone, "Enhancement of lithography patterns by using serif features," *IEEE Trans. Electron Devices*, vol. 38, no. 12, pp. 2599–2603, Dec. 1991.

[13] K. Harazaki, Y. hasegawa, Y. Shichijo, H. Tabuchi, and K. Fujii, "High Accurate Optical Proximity Correction under the Influences of Lens Aberration in 0.15 m Logic Process," *International Microprocesses and Nanotechnology Conference*, pp. 14–15, 2000.

[14] M. Hayashi and S. Tsukiyama, "A hybrid hierarchical global router for multi-layer VLSIs," *IEICE Trans. Fundamentals*, vol. E78-A, no. 3, pp. 337–344, 1995.

[15] J. Heisterman and T. Lengauer, "The efficient solutions of integer programs for hierarchical global routing," *IEEE Trans. CAD*, vol. 10, no. 6, pp. 748–753, June 1991.

[16] D. Hightower, "A solution to line routing problems on the continuous plane," *Proc. Design Automation Workshop*, pp. 1–24, 1969.

[17] T.-Y. Ho, Y.-W. Chang, S.-J. Chen, and D. T. Lee, "A Fast Crosstalk- and Performance-Driven Multilevel Routing System," *Proc. ICCAD*, pp.382–387, Nov. 2003.

[18] T.-Y. Ho, Y.-W. Chang, and S.-J. Chen, "Multilevel Routing with Antenna Avoidance," *Proc. ISPD*, pp. 34–40, April 2004.

[19] L.-D. Huang and D.-F. Wong, "Optical Proximity Correction (OPC)-Friendly Maze Routing," *Proc. DAC*, pp. 186–191, June 2004.

[20] G. Karypis, R. Aggarwal, V. Kumar, and S. shekhar, "Multilevel hypergraph partitioning: Application in VLSI domain," *IEEE Trans. VLSI Systems*, vol. 7, pp. 69–79, March 1999.

[21] R. Kastner, E. Bozorgzadeh and M. Sarrafzadeh, "Predictable routing," *Proc. ICCAD*, pp. 110–114, Nov. 2000.

[22] Lee, "An algorithm for path connection and its application," *IRE Trans. Electronic Computer*, EC-10, 1961.

[23] H.-C. Lee, Y.-W. Chang, J.-M. Hsu, and H. Yang, "Multilevel floorplanning/placement for large-scale modules using B*-trees," *Proc. DAC*, pp. 812–817, June 2003.

[24] Marc D. Levenson, N.S. Viswanathan, and Robert A. Simpson, "Improving Resolution in Photolithography with a Phase-Shifting Mask," *IEEE Trans. Electron Devices*, vol. 29, no. 12, pp. 1828–1836, Dec. 1982.

[25] Y. L. Lin, Y. C. Hsu, and F. S. Tsai, "Hybrid routing," *IEEE Trans. CAD*, vol. 9, no. 2, pp. 151–157, Feb. 1990.

[26] S. P. Lin and Y. W. Chang, "A novel framework for multilevel routing considering routability and performance," *Proc. ICCAD*, pp. 44–50, Nov. 2002.

[27] M. Marek-Sadowska, "Global Router for gate array," *Proc. ICCD*, pp. 332–337, Oct. 1984.

[28] M. Marek-Sadowska, "Router planner for custom chip design," *Proc. ICCAD*, Nov. 1986.

[29] G. Meixner and U. Lauther, "A new global router based on a flow model and linear assignment," *Proc. ICCAD*, pp. 44–47, Nov. 1990.

[30] Y.-C. Pati, Y.-T. Wnag, J.-W Liang, and Thomas Kailiath, "Phase-Shift Masks: Automated Design and Mask Requirements," *SPIE*, 2197:314–327, 1994.

[31] J. Soukup, "Fast maze router," *Proc. DAC*, pp. 100–102, June 1978.

[32] D. Wang and E. Kuh, "A new timing-driven multilayer MCM/IC routing algorithm," *Proc. Multi-chip Module Conference*, pp. 89–94, Feb. 1997.

[33] K. Yamamoto, S. Kobayashi, T. Uno, T. Kotani, S. Tanaka, S. Inoue, S. Watanabe, and H. Higurashi, "Hierarchical Optical Proximity Correction on contact Hole Layers," *International Microprocesses and Nanotechnology Conference*, pp. 40–41, 2000.