

# A Novel Framework for Multilevel Full-Chip Gridless Routing\*

Tai-Chen Chen  
Graduate Institute of Electronics Engineering  
National Taiwan University  
Taipei 106, Taiwan  
tcchen@eda.ee.ntu.edu.tw

Yao-Wen Chang  
Graduate Institute of Electronics Engineering  
and Department of Electrical Engineering  
National Taiwan University  
Taipei 106, Taiwan  
ywchang@cc.ee.ntu.edu.tw

Shyh-Chang Lin  
SpringSoft, Inc.  
Hsinchu 300, Taiwan  
chris@springsoft.com.tw

**Abstract**— Due to its great flexibility, gridless routing is desirable for nanometer circuit designs that use variable wire widths and spacings. Nevertheless, it is much more difficult than grid-based routing because of its larger solution space. In this paper, we present a novel “V-shaped” multilevel framework (called VMF) for full-chip gridless routing. Unlike the traditional “ $\Lambda$ -shaped” multilevel framework (inaccurately called the “V-cycle” framework in the literature), our VMF works in the V-shaped manner: top-down uncoarsening followed by bottom-up coarsening. Based on the novel framework, we develop a multilevel full-chip gridless router (called VMGR) for large-scale circuit designs. The top-down uncoarsening stage of VMGR starts from the coarsest regions and then processes down to finest ones level by level; at each level, it performs global pattern routing and detailed routing for local nets and then estimate the routing resource for the next level. Then, the bottom-up coarsening stage performs global maze routing and detailed routing to reroute failed connections and refine the solution level by level from the finest level to the coarsest one. We employ a dynamic congestion map to guide the global routing at all stages and propose a new cost function for congestion control. Experimental results show that VMGR achieves the best routability among all published gridless routers based on a set of commonly used MCNC benchmarks. Besides, VMGR can obtain significantly less wire-length, smaller critical path delay, and smaller average net delay than the previous works. In particular, VMF is general and thus can readily apply to other problems.

## I. INTRODUCTION

Routing complexity is an important problem for modern routers. To cope with the increasing complexity, the multilevel framework is proposed to solve the routing problems (e.g., MRS [8], MARS [9, 10], MR [3, 22], CMR [12, 14], MGR [4], XMR [15]) as well as graph/circuit partitioning (e.g., Chaco [11], ML [1], hMETIS [19], HPM [7]), floorplanning (e.g., MB\*-tree [21], MLGFA [16]), and placement (e.g., mPL [2] and APlace [17, 18]). All of the existing multilevel frameworks adopt a two-stage technique, bottom-up coarsening followed by top-down uncoarsening, which is known as the “ $\Lambda$ -shaped” framework. See Figure 1(a) for an illustration of the “ $\Lambda$ -shaped” multilevel routing framework. (Note that this framework is often called the “V-cycle” framework in the literature. However, we think that it is more appropriate to name it the “ $\Lambda$ -shaped” framework as it works bottom-up and then top-down.) These frameworks handle the target problems first bottom-up from local configurations to global ones and then refine the solutions top-down from global to local. It is obvious that there are significant limitations for the  $\Lambda$ -shaped framework to handle the global circuit effect, such as interconnection optimization, since only local information is available at the begin-

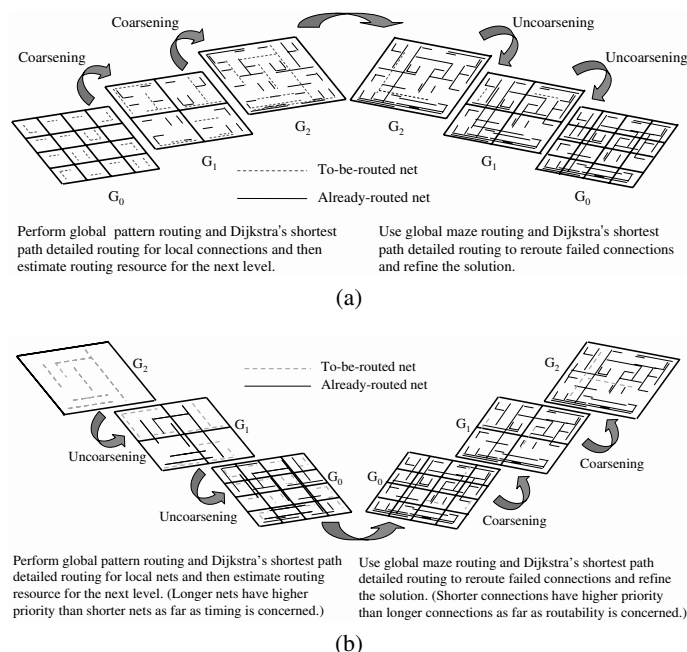


Fig. 1. (a) The  $\Lambda$ -shaped multilevel framework flow; (b) The V-shaped multilevel framework flow.

ning stages. A wrong choice made in such early stages may make the solution very hard to be refined during the top-down stage.

Most of the previous routing algorithms are grid-based, assuming uniform wire/via sizes. However, the grid-based approach is not effective to handle modern routing problems with nanometer electrical effects, such as optical proximity correction (OPC) and phase-shift mask (PSM). To cope with these nanometer electrical effects, we need to consider designs of variable wire/via widths and spacings, for which gridless routers are desirable due to their great flexibility. The gridless routing, however, 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. Cong et al. in [6] proposed a three-level routing scheme with a wire-planning phase between the global routing and the detailed routing. However, for large-scale designs, even with the three-level routing system, the problem size at each level may still be very large. Therefore, as the designs grow, more levels of routing are needed [10]. Recently, we proposed an OPC-aware multilevel gridless router based on the  $\Lambda$ -shaped framework [4], which integrates gridless global and detailed routing at each level. The router can handle non-uniform wire widths and reduce OPC pattern feature requirements.

In this paper, we present a new “V-shaped” multilevel routing framework (called VMF). Unlike the traditional  $\Lambda$ -shaped multilevel frameworks (called LMF) that apply bottom-up coarsening followed

\*This work was partially supported by SpringSoft, Inc. and National Science Council of Taiwan under Grant No's. NSC 94-2215-E-002-005 and NSC 94-2215-E-002-030.

Work	Category of routing	Framework	Characteristics
Ours	• Multilevel gridless global and detailed routing	<ul style="list-style-type: none"> <li>• Use V-shaped multilevel framework.</li> <li>• Before uncoarsening: channel density initialization.</li> <li>• Uncoarsening: GR+DR+RE.</li> <li>• Coarsening: global and detailed maze refinement.</li> </ul>	<ul style="list-style-type: none"> <li>• Perform global and detailed routing at each level.</li> <li>• Handle longer nets first and thus the wirelength and the critical path are reduced.</li> </ul>
Chang et al. in [3, 22]	• Multilevel grid-based global and detailed routing	<ul style="list-style-type: none"> <li>• Use <math>\Lambda</math>-shaped multilevel framework.</li> <li>• Coarsening: GR+DR+RE.</li> <li>• Uncoarsening: global and detailed maze refinement.</li> </ul>	<ul style="list-style-type: none"> <li>• Perform global and detailed routing at each level.</li> <li>• Lack initial global routing.</li> </ul>
Chen et al. in [4]	• Multilevel gridless global and detailed routing	<ul style="list-style-type: none"> <li>• Use <math>\Lambda</math>-shaped multilevel framework.</li> <li>• Coarsening: GR+DR+RE.</li> <li>• Uncoarsening: global and detailed maze refinement.</li> </ul>	<ul style="list-style-type: none"> <li>• Perform global and detailed routing at each level.</li> <li>• Lack initial global routing.</li> </ul>
Cong et al. in [8, 9, 10]	• Multilevel gridless global routing + flat gridless detailed routing	<ul style="list-style-type: none"> <li>• Use <math>\Lambda</math>-shaped multilevel framework.</li> <li>• Coarsening: RE.</li> <li>• Intermediate stage: multicommodity flow.</li> <li>• Uncoarsening: global maze refinement.</li> </ul>	<ul style="list-style-type: none"> <li>• Perform global and detailed routing separately.</li> </ul>
Ho et al. in [12, 13, 14]	• Multilevel grid-based global and detailed routing	<ul style="list-style-type: none"> <li>• Use <math>\Lambda</math>-shaped multilevel framework.</li> <li>• Coarsening: GR+RE.</li> <li>• Intermediate stage: track/layer assignment.</li> <li>• Uncoarsening: global and detailed maze refinement.</li> </ul>	<ul style="list-style-type: none"> <li>• Perform global and detailed routing separately.</li> </ul>

TABLE I

MULTILEVEL FRAMEWORK COMPARISONS AMONG [3, 22], [4], [8, 9, 10], [12, 13, 14], AND VMGR. GR, DR, AND RE DENOTE GLOBAL ROUTING, DETAILED ROUTING, AND RESOURCE ESTIMATION, RESPECTIVELY.

by top-down uncoarsening, VMF adopts the two-stage technique of top-down uncoarsening followed by bottom-up coarsening. See Figure 1(b) for an illustration of VMF. The V-shaped multilevel framework was first introduced for interconnect-driven floorplanning [5]; it outperforms the  $\Lambda$ -shaped one in optimizing global circuit effects (such as wirelength, timing, and crosstalk optimization), since the V-shaped framework first considers the global configuration and then processes down to local ones level by level and thus the global effects can be handled at earlier stages.

Based on VMF, we develop a V-shaped multilevel full-chip gridless router (called VMGR) for large-scale circuit designs. The top-down uncoarsening stage of VMGR starts from the coarsest regions and then processes down to finest ones level by level; at each level, it performs global pattern routing and detailed routing for local nets and then estimate the routing resource for the next level. Then, the bottom-up coarsening stage performs global maze routing and detailed routing to reroute failed connections and refine the solution level by level from the finest level to the coarsest one.

In addition to the aforementioned characteristics, our VMF-based VMGR has the following distinguished features:

- The previous works [3, 12, 13, 14, 22] are *grid-based* multilevel router, which cannot handle designs of variable wire/via widths and spacings. Thus, they cannot effectively handle modern routing problems with nanometer electrical effects such as OPC.
- VMF considers the global longer nets first at the earlier uncoarsening stage, leading to better control on critical path delay and global interconnect effects.
- The previous works [3, 4, 22] perform *greedy global routing*, which determines the global path of the current net without considering the routing resource of succeeding nets. In contrast, VMGR employs a congestion map to guide the global routing at all stage. Initially, the map keeps the preliminary estimation of routing congestion based on the pin distribution. After routing a net, the map is updated dynamically based on the real route, previously routed nets, and estimated unrouted nets. As routing proceeds, we keep more and more accurate congestion information in the map. Therefore, we have better congestion control throughout the whole routing process.
- We use a new cost function based on *both* the total path congestion and the maximum channel congestion for global routing.

The cost function obtains better solutions than those consider only total path congestion or the maximum channel congestion.

- VMGR has higher flexibility and keeps more global views, and thus more routing objectives (such as crosstalk and OPC) can be more easily considered in VMGR since exact track and wiring information at each level after detailed routing is known.

Table I compares the existing multilevel routing frameworks among [3, 22], [4], [8, 9, 10], [12, 13, 14], and VMF.

Experimental results show that our VMGR achieves the best routability among all published gridless routers [4, 10] based on a set of commonly used MCNC benchmarks with non-uniform and uniform wire widths.

The rest of this paper is organized as follows. Section II presents the global, detailed, and V-shaped multilevel routing models. Section III presents our V-shaped multilevel routing framework. Experimental results are reported in Section IV. Finally, we give concluding remarks in Section V.

## II. PRELIMINARIES

Routing in modern IC's is a very complex process, and we can hardly obtain high-quality solutions directly. Therefore, the routing problem is usually solved using the two-stage approach of global routing followed by detailed routing. Global routing first partitions the routing area into tiles and decides tile-to-tile paths for all nets while detailed routing assigns actual tracks and vias for nets.

### A. Modeling of Global Routing

Our global routing algorithm is based on a graph search technique guided by the congestion 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. Fig. 2 illustrates the graph modeling. For the modeling, we first partition a chip into an array of rectangular subregions. These subregions are called *global routing cells (GRCs)*. A node in the routing graph represents a *GRC*

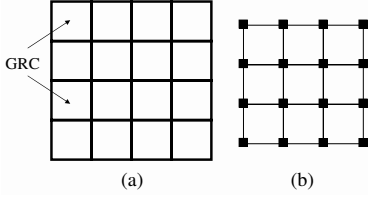


Fig. 2. Modeling of global routing: (a) Partitioned layout; (b) Routing graph.

in the chip, and an edge denotes the boundary between two adjacent *GRCs*. Each edge is assigned a capacity according to the width/height of a *GRC*. The routing graph is used to represent the routing area and is called a *multilevel routing graph*, denoted by  $G_k$ , where  $k$  is the level ID. A global router finds *GRC*-to-*GRC* 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.

### B. Modeling of Detailed Routing

In the detailed routing stage, seeking high-quality and design-rule-correct paths in the routing region are two major concern. A suitable detailed routing model greatly affects these concerns. At first, for each obstacle, its obstacle zone is constructed by expanding the obstacle for a range which is the sum of the obstacle spacing and the half width of the routing wire. As shown in Fig. 3(a), the expanded range (gray area) is the sum of  $D_S$  and  $W_i/2$ , where  $D_S$  and  $W_i$  are the obstacle spacing to satisfy the design rules and the width of the routing wire, respectively. With the boundaries of each all extended regions and the center of the obstacle zone, three  $x$ -coordinates (the left boundary, the right boundary, and the center) and three  $y$ -coordinates (the top boundary, the bottom boundary, and the center) are obtained. The  $x$ -coordinates and  $y$ -coordinates of all obstacle zones and the source  $P_S$  and target  $P_T$  of the routing wire are stored into two sets,  $ICG_x$  and  $ICG_y$ , separately. Based on  $ICG_x$  and  $ICG_y$ , an implicit connection graph is constructed, as shown in Fig. 3(b). A vertical (horizontal) dashed lines in the implicit connection graph is generated through each  $x$ -coordinate ( $y$ -coordinate) in  $ICG_x$  ( $ICG_y$ ). A node in the implicit connection graph denotes an intersection of a horizontal and a vertical dashed lines. There are two types of nodes, *routable nodes* and *unroutable nodes*. A routable node allows a routing path to pass through it without violating the design rules; it is unroutable, otherwise. As shown in Fig. 3(b), the respective black and white circles are the routable and unroutable nodes. To seek a design-rule-correct path from the source  $P_S$  to the target  $P_T$ , therefore, we only need to check if there exists a feasible path along which all nodes are routable. As shown in Fig. 3(c), a design-rule-correct path from  $P_S$  to  $P_T$  is found through the eleven routable nodes.

### C. Modeling of V-Shaped Multilevel Routing

As illustrated in Figure 1(b),  $G_0$  corresponds to the routing graph of the level 0 of the multilevel uncoarsening stage. Before the uncoarsening stage is performed, we need to determine the number of levels and build *GRCs* for each level. For each level  $i$ , we merge four  $GRC_i$  of  $G_i$  into a larger  $GRC_{i+1}$ . The process continues until the number of *GRCs* at level  $k$  is equal to one. Note that this process is just a pre-processing for determining  $k$  and no any global routing, detailed routing, or resource estimation is involved. Therefore, the pre-processing is different from the coarsening. After determining the number of levels, we start with the uncoarsening stage from the

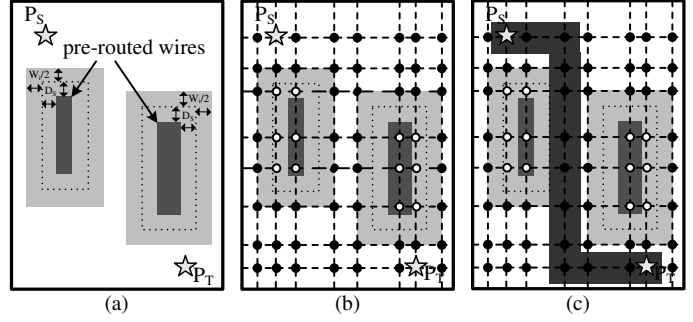


Fig. 3. (a) A routing example. The gray areas denote the obstacle zones which are constructed by expanding a range which is the sum of the wire spacing and the half width of the routing wire.  $D_S$  and  $W_i$  are wire/via spacing and the width of the routing wire that satisfy the design rules, respectively.  $P_S$  and  $P_T$  are the source and target of the routing wire, respectively. (b) The implicit connection graph constructed by our detailed model. (c) A design-rule-correct path found through the eleven routable nodes.

$k$ -th level. At each level  $i$ , our global router just finds routing paths for the *local nets* (a local net at level  $i$  denotes that all pins of the net can be included entirely by a  $GRC_i$  and cannot be included totally by a  $GRC_{i-1}$ ), and then the detailed router is used to determine the exact wiring. After the global and detailed routing are performed, we expand each  $GRC_i$  to four finer  $GRC_{i-1}$  and at the same time perform resource estimation. The uncoarsening stage continues until the 0-th level is arrived. After finishing the uncoarsening stage, the coarsening stage tries to refine the routing solution starting from the level 0. During the coarsening stage, the unroutable connections during the uncoarsening stage are considered, and point-to-path maze routing and rip-up and re-route are performed to refine the routing solution. Then we proceed to the next level (i.e., level 1 here) of the coarsening stage by merging four adjacent  $GRC_0$  into a larger  $GRC_1$ . The process continues until we go back to level  $k$  when the final routing solution is obtained.

## III. V-SHAPED MULTILEVEL ROUTING FRAMEWORK

VMGR tends to route wider nets first since a wider net consumes more routing resource. Besides, VMGR tends to route longer nets first at the uncoarsening stage. It is obvious that the local nets at the higher level (say, level  $k$ ) are usually longer than those at a lower level (say, level 0). Usually, a longer net has larger path delay. Thus, this observation implicitly suggests that a longer net has a higher priority than a shorter net as far as timing is concerned. Though this net ordering scheme may not be the optimal solution for some routing problems (for example, when routability is considered, routing shorter nets first often leads to a better completion rate), it is still a better alternative to the optimization of global interconnect effects.

### A. Channel Density Initialization and Update

If global routing, detailed routing, and resource estimation are performed separately, the re-routing process conducted at the global routing stage may be in vain since it does not know if the re-routing is useful for the detailed routing. Also, the detailed router may fail to find a path because of the low flexibility induced from the separated global routing. Therefore, making the three tasks interact with each other can significantly improve routing quality [3, 22]. However, the concept can only guide the latter nets passing through the area with lower congestion and cannot avoid a wrong decision made by *greedy global routing* which determines the global path of an early routed net

without considering the routing resource of succeeding nets. Therefore, we initialize the routing congestion information based on the pin distribution and the global-path prediction of all nets, and then keep a congestion map that is updated dynamically based on both the already routed nets and the estimated unrouted nets. As routing proceeds, we keep more and more accurate congestion information in the map. Therefore, we have better congestion control throughout the whole routing process.

For a 2-pin connection  $c$ , we use L- and Z-shaped routes to determine the number of possible global routes  $n_c$ . We evenly distribute the wire density of the connection  $c$ ,  $w_c$ , among all possible global routes. Therefore, the wire density of each possible global route is  $w_c/n_c$ . For each possible global route, we add the wire density of the possible global route to the channel density in the routing graph. After all 2-pin connections finish the process, we get an initial channel density. Note that the aforementioned approach is a natural way to estimate routing congestion, commonly used for interconnect-driven floorplanning.

At first, the channel density is totally estimated by the approach. After a connection has been routed successfully, the estimated cost induced by the connection will be removed from the channel density, and the wire density of the real path will be updated to the channel density (congestion map) dynamically. Therefore, our congestion control is based on congestion information induced by both the already routed nets and the estimated unrouted nets. As routing proceeds, we have more and more accurate congestion information for routing succeeding nets.

### B. Cost Function for Global Routing

Let the multilevel routing graph be  $G_0 = (V_0, E_0)$ . Let  $R_e = \{e \in E_0 \mid e \text{ is the edge chosen for routing}\}$ . We apply the cost function  $\alpha : E_0 \rightarrow \Re$  to guide the global routing:

$$\alpha(R_e) = \max_{e \in R_e} c_e + \frac{1}{|R_e|} \sum_{e \in R_e} c_e, \quad (1)$$

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

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

where  $p_e$  and  $d_e$  are the capacity and channel density associated with  $e$ , respectively. 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.)

There are two advantages by using this cost function for global routing. First, this cost function can avoid that we select a path which has lower total path congestion with a higher channel congestion. Second, this cost function can prevent us from choosing a worse global path with the higher overall path congestion when two global paths have the same maximum channel congestion.

### C. V-shaped Multilevel Gridless Routing

In the following, we present our framework for VMGR and summarize it in Figure 4.

Given a netlist, we first run a 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 MST. According to those 2-pin connections, we use the heuristic in Section A to initialize the channel density in the routing graph by predicting the global paths of all nets in advance.

```

Algorithm: V-shaped-Multilevel-Gridless-Routing( $G, N$ )
  Input:  $G$  - partitioned layout;
            $N$  - netlist of multi-terminal nets;
  Output: routing solutions for  $N$  on  $G$ 
begin
1  Partition layout;
2  For each net  $n \in N$ 
3    Construct an MST;
4    Decompose the MST into 2-pin connections;
5    For each 2-pin connection
6      Initialize channel density;
7  // Uncoarsening Stage
8  For each level at the uncoarsening stage
9    Choose a local net  $n$ ;
10   For each connection  $c \in n$ ;
11     Perform global pattern routing;
12     Perform detailed routing;
13     Update channel density;
14 // Coarsening Stage
15 For each level at the coarsening stage
16   Choose a failed connection at the uncoarsening stage
17   Perform global maze routing;
18   Perform detailed routing;
19   Update channel density;
20 Analyze timing for all nets;
21 return the routing layout;
end

```

Fig. 4. Algorithm for V-shaped multilevel gridless routing.

VMGR starts from uncoarsening the coarsest tile of level  $k$ . At each level, tiles are processed one by one, and only local nets 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 [20] and first routes local nets on the tiles of level  $k$ . Let the multilevel routing graph of level  $i$  be  $G_i = (V_i, E_i)$ . Let  $R_e = \{e \in E_i \mid e \text{ is the edge chosen for routing}\}$ . We apply the cost function in Section B to guide the routing.

After the global routing is completed, VMGR performs detailed routing with the guidance of the global-routing results and finds a real path in the chip. Our detailed router is based on the Dijkstra's shortest path algorithm and supports the *local refinement*. If detailed routing of a connection fails, it will be reconsidered (refined) at the coarsening stage. After a connection has been routed successfully, the estimated cost induced by the connection which calculated by the approach in Section A will be removed from the channel density, and the wire density of the real path will be updated to the channel density (congestion map) dynamically. This is called *resource estimation*. There are at least two advantages by using this approach. First, routing resource 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 coarsening stage starts to refine each local failed connection, left from the uncoarsening stage. The global router is now changed to the maze router with the same cost function in the uncoarsening stage. Coarsening continues until the first level  $k$  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 uncoarsening. (For rip-up and re-route in VMGR, we mean the maze routing at the coarsening 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.

#### IV. EXPERIMENTAL RESULTS

We implemented VMGR in the C++ language on a 1 GHz SUN Blade-2000 workstation with 8 GB memory. We compared our results with the gridless routers presented in [4, 10] based on the 11 benchmark circuits provided by the authors. (Note that since the results of [10] is better than those of [8, 9], we just compare our results with [10].) The design rules for wire/via widths and wire/via spacings for detailed routing are the same as those used in [10].

Table II lists the set of benchmark circuits. In the table, “Circuit” gives the names of the circuits, “Size ( $\mu m^2$ )” gives the layout dimensions in  $\mu m^2$ , “#Layers” denotes the number of routing layers used, “#Nets” gives the number of two-pin connections after net decomposition, and “#Pins” gives the number of pins. For delay computation, we use the Elmore delay model. All the parameters are the same as those used in [4]. A via is modeled as the  $\Pi$ -model circuit, with its resistance and capacitance being twice of those of a wire segment. As pointed out in [3, 22], Mcc1, Mcc2, Struct, Primary1, and Primary2 do not have the information of net sources. Therefore, we cannot calculate the path delay for those benchmark circuits. In the following experiments, we represent the critical path and average net delays of these 5 benchmark circuits by the notation, –.

Circuit	Size ( $\mu m^2$ )	#Layers	#Nets	#Pins
Mcc1	45000 × 39000	4	1693	3101
Mcc2	152400 × 152400	4	7541	25024
Struct	4903 × 4904	3	3551	5471
Primary1	7522 × 4988	3	2037	2941
Primary2	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 II  
THE BENCHMARK CIRCUITS.

##### A. Multilevel Gridless Routing with Uniform Nets

Table III lists the experimental results obtained by the  $\Lambda$ -shaped multilevel gridless routing in [4] (called LMGR), the  $\Lambda$ -shaped multilevel gridless routing system (multilevel global routing + flat gridless detailed routing) in [10] (called MARS), and VMGR. In the table, “WL ( $\mu m$ )” represents the wirelength in  $\mu m$ , “ $D_{max}$  (psec)” represents the critical path delay in pico-second, “ $D_{avg}$  (fsec)” represents the average net delay in femto-second, “Comp. Rates” gives the routing completion rates, and “Time (sec)” represents the runtime in second.

Compared with LMGR, the experimental results show that VMGR achieves a 5.19X runtime speedup while LMGR results in longer wirelength, larger critical path delay, and larger average net delay (1.02X wirelength, 1.21X critical path delay, and 1.00X average net delay). Compared with MARS, the experimental results show that VMGR achieves a 1.97X runtime speedup. (Note that it is hard to make a fair comparison between MARS and VMGR, because MARS and VMGR ran on different machines. Nevertheless, they both ran on

SUN workstations. Therefore, we try our best to make a fair comparison by normalizing the runtime based their clock rates.) Since MARS did not report their wirelength, critical path delay, and average net delay in their paper, we cannot compare those results in MARS with VMGR.

##### B. Multilevel Gridless Routing with Non-Uniform Nets

We also performed experiments on the benchmark circuits of non-uniform wire widths. We modify the original circuits of uniform wire sizes to generate a set of circuits of non-uniform wire sizes by using the following rules, which was proposed by [10]. The longest 10% nets are widened to twice the original width, while the next 10% are widened to 150% the original width. However, because the benchmark circuits S5378–S38584 are standard-cell designs, widening any pin violates the design rules for via spacing. Therefore, it is unreasonable and incorrect to test these six modified benchmark circuits.

In Table IV, “#Total Sub-nets” denotes the total number of 2-pin nets seen by the detailed router of MARS, since the detailed router of MARS segments long two-pin nets into short subnets. As shown in the table, VMGR still achieves 100% routing completion for all of the 5 circuits with 1.91X (1.19X) runtime speedup while [4] ([10]) completes routing for only 4 circuits. Note that VMGR is the first router to complete the routing for this set of benchmarks of non-uniform wire sizes. In particular, we expect that the difference will be much more significant for larger and difficult designs such as vd\_Mcc2. Figures 5 and 6 show the full-chip and partial routing solutions for “vd\_Mcc2” obtained from VMGR, respectively. The bounding box in Figure 5 is the boundary of this benchmark circuit. We can see in Figure 6 that the three left-most vertical lines have different widths.

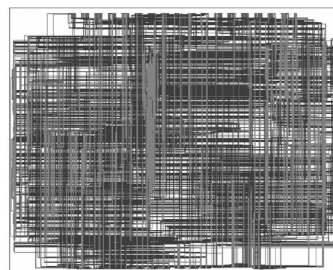


Fig. 5. The full-chip routing solution for “vd\_Mcc2” obtained from VMGR. The bounding box is the boundary of this benchmark circuit.

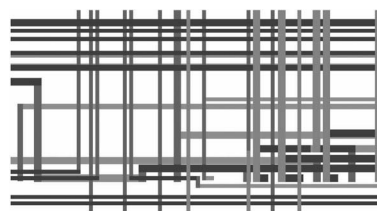


Fig. 6. A partial routing solution for “vd\_Mcc2” obtained from VMGR. We can see that the three left-most vertical lines have different widths.

The two experimental results reveal the effectiveness of VMF for multilevel routing. Since VMF considers the global longer nets first at the earlier uncoarsening stage, it can have better control on the wirelength and the critical path delay. Besides, the runtime and solution quality are improved simultaneously. Also, compared with [4] that was based on LMF, the experimental results have shown that LMF leads to significantly better wirelength, critical path delay, and average net delay, and 100% routing completion rates.

Circuit	(A) Results of [4]					(B) Results of [10]					(C) Our Results				
	WL ( $\mu\text{m}$ )	$D_{max}$ (psec)	$D_{avg}$ (fsec)	Comp. Rates	Time (sec)	WL ( $\mu\text{m}$ )	$D_{max}$ (psec)	$D_{avg}$ (fsec)	Comp. Rates	Time (sec)	WL ( $\mu\text{m}$ )	$D_{max}$ (psec)	$D_{avg}$ (fsec)	Comp. Rates	Time (sec)
Mcc1	2.8e7	-	-	100%	190.2	NA	-	-	100%	105.1	2.7e7	-	-	100%	56.4
Mcc2	4.1e8	-	-	100%	3711.0	NA	-	-	100%	1916.9	4.0e8	-	-	100%	1353.8
Struct	8.5e5	-	-	100%	6.5	NA	-	-	100%	31.6	8.4e5	-	-	100%	4.4
Primary1	1.0e6	-	-	100%	5.1	NA	-	-	100%	33.5	1.0e6	-	-	100%	4.7
Primary2	4.2e6	-	-	100%	46.7	NA	-	-	100%	162.7	4.1e6	-	-	100%	27.5
S5378	7.6e4	21	780	100%	45.6	NA	NA	NA	100%	30.0	7.4e4	11	777	100%	5.7
S9234	5.5e4	18	681	100%	25.1	NA	NA	NA	100%	22.8	5.4e4	17	678	100%	4.3
S13207	1.8e5	37	828	100%	136.2	NA	NA	NA	100%	85.2	1.8e5	33	812	100%	17.9
S15850	2.2e5	87	855	100%	362.2	NA	NA	NA	100%	107.1	2.2e5	84	866	100%	22.7
S38417	4.8e5	183	759	100%	403.1	NA	NA	NA	100%	250.9	4.7e5	174	763	100%	70.7
S38584	6.7e5	1086	835	100%	765.1	NA	NA	NA	100%	466.1	6.6e5	1026	828	100%	209.0
Comp.	1.02	1.21	1.00	1	5.19				1	1.97*	1	1	1	1	1

TABLE III

COMPARISON AMONG (A) THE  $\Lambda$ -SHAPED MULTILEVEL GRIDLESS ROUTING [4], (B) THE  $\Lambda$ -SHAPED MULTILEVEL GRIDLESS GLOBAL ROUTING + FLAT GRIDLESS DETAILED ROUTING [10], AND (C) VMGR. NOTE: (A) AND (C) RAN ON A 1 GHZ SUN BLADE-2000 WITH 8 GB MEMORY; (B) RAN ON A 440 MHZ SUN ULTRA-10 WITH 384 MB MEMORY. (-: BECAUSE THOSE BENCHMARK CIRCUITS DID NOT HAVE THE INFORMATION OF NET SOURCES, WE CANNOT CALCULATE THE PATH DELAY FOR THEM.) (NA: [10] DID NOT REPORT THEIR WIRELENGTH, CRITICAL PATH DELAY, AND AVERAGE NET DELAY IN THEIR PAPER.) (\*: FOR FAIR COMPARISONS, WE NORMALIZE THE RUNTIME OF [10] BY THE FACTOR 440/1000.)

Circuit	(A) Results of [4]				(B) Results of [10]				(C) Our Results			
	WL ( $\mu\text{m}$ )	#Failed Nets	Comp. Rates	Time (sec)	WL ( $\mu\text{m}$ )	#Failed Nets (#Total Sub-nets)	Comp. Rates	Time (sec)	WL ( $\mu\text{m}$ )	#Failed Nets	Comp. Rates	Time (sec)
vd_Mcc1	2.8e7	0	100%	199.6	NA	0	100%	148.1	2.7e7	0	100%	65.4
vd_Mcc2	4.1e8	383	98.5%	36581.5	NA	27(99715)	99.97%	3388.8	4.1e8	0	100%	23383.3
vd_Struct	8.5e5	0	100%	15.3	NA	0	100%	36.3	8.4e5	0	100%	10.3
vd_Primary1	1.0e6	0	100%	19.2	NA	0	100%	47.4	1.0e6	0	100%	12.2
vd_Primary2	4.2e6	0	100%	150.8	NA	0	100%	296.7	4.1e6	0	100%	80.0
Comp.	1.02		99.70%	1.91			99.99%	1.19*	1		1	1

TABLE IV

COMPARISON AMONG (A) THE  $\Lambda$ -SHAPED MULTILEVEL GRIDLESS ROUTING [4], (B) THE  $\Lambda$ -SHAPED MULTILEVEL GRIDLESS GLOBAL ROUTING + FLAT GRIDLESS DETAILED ROUTING [10], AND (C) VMGR. NOTE: (A) AND (C) RAN ON A 1 GHZ SUN BLADE-2000 WITH 8 GB MEMORY; (B) RAN ON A 440 MHZ SUN ULTRA-10 WITH 384 MB MEMORY. (NOTE THAT BECAUSE THE BENCHMARK CIRCUITS S5378–S38584 VIOLATE THE DESIGN RULES OF VIA SPACING, WE DID NOT LIST THESE CASES IN THIS TABLE.) (NA: [10] DID NOT REPORT THEIR WIRELENGTH IN THEIR PAPER.) (\*: FOR FAIR COMPARISONS, WE NORMALIZE THE RUNTIME OF [10] BY THE FACTOR 440/1000.)

## V. CONCLUSION

In this paper, we have proposed a novel V-shaped framework for multilevel, full-chip gridless routing. The V-shaped multilevel framework adopts a two-stage technique, top-down uncoarsening followed by bottom-up coarsening. Experimental results have shown that our V-shaped multilevel gridless router can obtain 100% routing completion rates with less wirelength, smaller critical path delay, and smaller average net delay than previous works. Besides, it can handle designs with non-uniform wire widths well and obtained better routing solutions than previous works. In particular, our gridless router is the first to complete the routing for the set of commonly used benchmarks of non-uniform wire sizes listed in the preceding section.

## REFERENCES

- [1] 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.
- [2] T. Chan, J. Cong, T. Kong, and J. Shinnerl, "Multilevel optimization for large-scale circuit placement," *Proc. ICCAD*, pp. 171–176, Nov. 2000.
- [3] 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.
- [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, "IMF: Interconnect-driven multilevel floorplanning for large-scale building-module designs," *Proc. ICCAD*, pp. 159–164, Nov. 2005.
- [6] J. Cong, J. Fang, and K. Khoo, "DUNE: A multi-layer gridless routing system with wire planning," *Proc. ISPD*, pp. 12–18, April 2000.
- [7] J. Cong, S. Lim, and C. Wu, "Performance driven multilevel and multiway partitioning with retiming," *Proc. DAC*, pp. 274–279, June 2000.
- [8] J. Cong, J. Fang, and Y. Zhang, "Multilevel approach to full-chip gridless routing," *Proc. ICCAD*, pp. 396–403, Nov. 2001.
- [9] J. Cong, M. Xie, and Y. Zhang, "An enhanced multilevel routing system," *Proc. ICCAD*, pp. 51–58, Nov. 2002.
- [10] J. Cong, J. Fang, M. Xie, and Y. Zhang, "MARS—A multilevel full-chip gridless routing system," *IEEE Trans. CAD*, vol. 24, no. 3, pp. 382–394, March 2005.
- [11] B. Hendrickson and R. Leland, "A multilevel algorithm for partitioning graph," *Proc. Supercomputing*, pp. 1–24, July 1995.
- [12] 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.
- [13] T.-Y. Ho, Y.-W. Chang, and S.-J. Chen, "Multilevel routing with antenna avoidance," *Proc. ISPD*, pp. 34–40, April 2004.
- [14] T.-Y. Ho, Y.-W. Chang, S.-J. Chen, and D.-T. Lee, "Crosstalk- and performance-driven multilevel full-chip routing," *IEEE Trans. CAD*, vol. 24, no. 6, pp. 869–878, June 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, June 2003.
- [16] C.-C. Hu, D.-S. Chen, and Y.-W. Wang, "Fast multilevel floorplanning for large scale modules," *Proc. ISCAS*, pp. 205–208, May 2004.
- [17] A.B. Kahng and Q. Wang, "Implementation and extensibility of an analytic placer," *Proc. ISPD*, pp. 18–25, April 2004.
- [18] A.B. Kahng, S. Reda, and Q. Wang, "APlace: A general analytic placement framework," *Proc. ISPD*, pp. 233–235, April 2005.
- [19] 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.
- [20] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "Predictable routing," *Proc. ICCAD*, pp. 110–114, Nov. 2000.
- [21] 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.
- [22] S.-P. Lin and Y.-W. Chang, "A novel framework for multilevel routing considering routability and performance," *Proc. ICCAD*, pp. 44–50, Nov. 2002.