

# An Integer-Linear-Programming-Based Routing Algorithm for Flip-Chip Designs

Jia-Wei Fang, *Student Member, IEEE*, Chin-Hsiung Hsu, *Student Member, IEEE*, and Yao-Wen Chang, *Member, IEEE*

**Abstract**—The flip-chip package provides a high chip-density solution to the demand for more input-output pads of very large scale integration designs. In this paper, we present the first routing algorithm in the literature for the preassignment flip-chip routing problem with a predefined netlist among pads and wire-width and signal-skew considerations. Our algorithm is based on integer linear programming (ILP) and guarantees to find an optimal solution for the addressed problem. It adopts a two-stage technique of global routing followed by detailed routing. In global routing, it first uses three reduction techniques to prune redundant solutions and create a global-routing path for each net. Without loss of the solution optimality, our reduction techniques can further prune the ILP variables (constraints) by 85.5% (98.0%) on average over a recent reduction technique. The detailed routing applies passing-point assignment, net-ordering determination, and X-based gridless routing to complete the routing. Experimental results based on five real industry designs show that our router can achieve 100% routability and the optimal global-routing wire-length, and satisfy all signal-skew constraints, under reasonable central-processing-unit times, whereas recent related work has resulted in much inferior solution quality.

**Index Terms**—Detailed routing, global routing, layout, physical design.

## I. INTRODUCTION

THE increasing complexity and the decreasing feature size of very large scale integration (VLSI) designs make the demand of more I/O pads a significant problem to package technologies. An advanced packaging technology, the *flip-chip (FC) package*, as shown in Fig. 1(a), is created for higher integration density, rising power consumption, and larger I/O counts.

FC is not a specific package, or even a package type, e.g., pin grid array (PGA) or ball grid array (BGA). FC describes the

Manuscript received November 26, 2007; revised July 15, 2008. Current version published December 17, 2008. This work was supported in part by the Etron, the SpringSoft, the Taiwan Semiconductor Manufacturing Company, and the National Science Council of Taiwan under Grant NSC 96-2628-E-002-248-MY3, Grant NSC 96-2628-E-002-249-MY3, Grant NSC 96-2221-E-002-245, Grant NSC 96-2752-E-002-008-PAE, and Grant NSC 96-2917-I-002-120. An earlier version of this paper was nominated for the Best Paper Award at the ACM/IEEE Design Automation Conference (DAC'07), San Diego, CA, June 2007 [7]. This paper was recommended by Associate Editor L. Scheffer.

J.-W. Fang and C.-H. Hsu are with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: jiawei@eda.ee.ntu.edu.tw; arious@eda.ee.ntu.edu.tw).

Y.-W. Chang is with the Department of Electrical Engineering and Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: ywchang@cc.ee.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2008.2009151

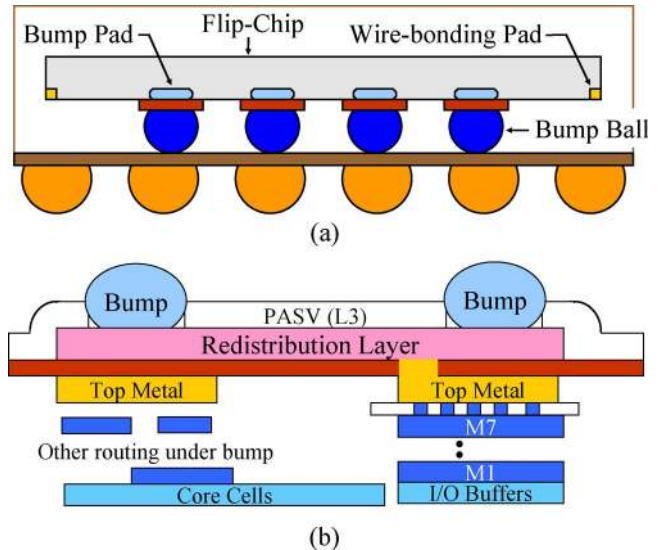


Fig. 1. (a) FC package. (b) Cross section of the RDL.

method of electrically connecting the die to the package carrier. FC technology is the choice in high-speed applications because of the following advantages: reduced signal inductance, reduced power consumption, reduced package footprint, higher signal density, etc. However, in recent IC designs, the I/O pads are still placed along the boundary of a die. This placement does not suit the FC package well. As a result, the top metal or an extra metal layer, called a *redistribution layer (RDL)*, as shown in Fig. 1(b), is used to redistribute the *wire-bonding pads* to the *bump pads* without changing the placement of the I/O pads. Since the RDL is the top metal layer of a die, the routing in an RDL cannot be any angle as in the PGA/BGA packages. Bump balls are placed on the RDL and use the RDL to connect to wire-bonding pads by bump pads. Therefore, a special router, the *RDL router*, is needed to reroute the peripheral wire-bonding pads to the bump pads (balls) [17]. In addition to the traditional routing cost metric of total wirelength, the issue of signal skews is of significant importance because the FC design is typically for high-speed circuits. Furthermore, multipin nets and variable wire widths are also significant considerations for an RDL router.

There are two kinds of the RDL routing problems for the FC design. The first one is the *free-assignment* routing problem. In this problem, a wire-bonding pad is not assigned to any bump pad before routing. Therefore, a router has the freedom to assign a wire-bonding pad to the bump pads during routing. Since the netlist is defined by the router, this routing problem

is relatively easier and can be solved by a network-flow formulation [6], [8]. The second kind of RDL routing is the *preassignment* routing problem, where the mappings among the wire-bonding pads and the bump pads are defined before routing and cannot be changed. Since the preassignment of the netlist imposes more routing constraints, this problem is much harder than the free-assignment one. Furthermore, the preassignment problem is, in fact, more popular in practice since the functions of wire-bonding and bump pads are typically predefined by IC and packaging designers. To the best of our knowledge, however, there is no existing work on the preassignment RDL routing for FC designs in the literature, and designers are often forced to awkwardly apply existing routers that are designed for other technologies to handle the RDL routing [17]. Obviously, this is by no means a right way to handle this problem. Therefore, it is greatly desired to develop an effective and efficient algorithm for RDL routing.

### A. Previous Work

As just mentioned, there is no previous work in the literature on the preassignment routing problem for FC designs. Recently, Fang *et al.* [6], [8] have addressed the free-assignment routing problem and presented a network-flow algorithm to assign wire-bonding pads to bump pads. Since the network-flow approach cannot guarantee the correct connections between two designated nodes, it cannot handle the preassignment routing problem. Other related works include the routing for planar graphs [1], [2], [12], PGA packages [3], [16], and BGA packages [4], [11], [15]. For [1], [2], and [12] on the planar routing, since modules with pins can be placed anywhere in a chip, it is harder and has been shown to be NP-complete, and, thus, most likely, there exists no efficient optimal algorithm for the planar routing. In the FC routing, since wire-bonding pads and bump pads are placed in arrays, we can take the advantage of the regular structure to find an efficient algorithm for the RDL routing. Thus, the FC routing problem is also different from the planar routing one. References [3] and [16] presented PGA routers, whereas [4] and [11] provided BGA routers. These routers are any-angle multilayer routers and do not consider the single-layer restriction, the signal skews, the variable wire widths, and the total wirelength minimization. Therefore, they are not suitable for the RDL routing, typically with a routing angle of  $90^\circ$  or  $45^\circ$ , and much more stringent performance constraints. A recent work [15] has provided an any-angle and single-layer BGA *global* router. It used an order graph to get a routing sequence for the nets. Since the relation of the nets is not all kept in the order graph, the number of legal routing sequences is typically huge. Furthermore, the graph does not keep the information of routing resources. As a result, it may not complete the routing even with a legal routing sequence. Also, this paper did not consider signal skews, variable wire widths, U-turn routes, and total wirelength minimization.

### B. Our Contributions

In this paper, we present the *first* algorithm for the preassignment RDL routing problem, considering signal skews, variable

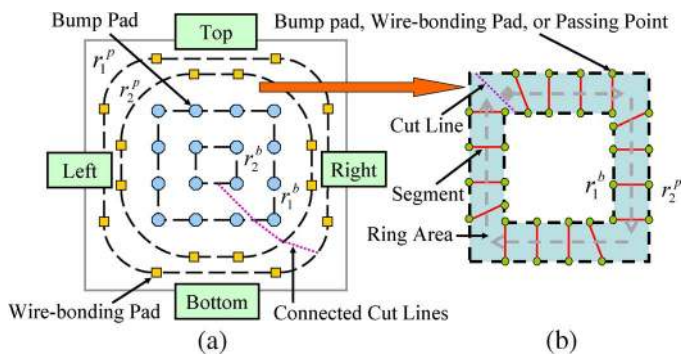


Fig. 2. (a) Four sides in an FC. (b) Cut lines, segments, and a ring area.

wire widths, U-turn routes, and total wirelength minimization. Our algorithm is based on integer linear programming (ILP) and guarantees to find an optimal solution for the addressed problem. It adopts a two-stage technique of global routing followed by detailed routing. In global routing, it simultaneously determines the routing sequence for the nets and creates global-routing paths among wire-bonding pads and bump pads. Formulating the global routing as ILP with the routing resource consideration, we can guarantee 100% detailed routing completion after global routing. Since ILP is NP-complete [9], it is computationally expensive. We apply three reduction techniques to prune redundant solutions and, thus, speed up the computation. Without loss of the solution optimality, our reduction techniques can further prune the ILP variables (constraints) by 85.5% (98.0%) on average over the order-graph technique presented in [15]. Due to the significant reductions and the optimality guarantee, our ILP-based algorithm is able to find very high quality solutions in reasonable CPU times. The detailed routing consists of three steps:

- 1) passing-point assignment to distribute the routing points between two adjacent wire-bonding (bump) pads according to the wire width and the skew constraint of each net;
- 2) net-ordering determination to identify better routing sequences;
- 3) X-based gridless routing to complete the routing with shorter wirelength (than Manhattan routing).

Experimental results based on five real industry designs show that our router can achieve 100% detailed-routing completion and the optimal global-routing wirelength, and satisfy all signal-skew constraints, under reasonable CPU times, compared to an integrated router (the BGA global router in [15] followed by the detailed router in [6] and [8]) that results in only 82.9% routability, all signal-skew violations, and 15.56% longer wirelength.

The rest of this paper is organized as follows. Section II gives the formulation of the RDL routing problem. Section III details our global and detailed routing algorithms. Section IV reports the experimental results. Finally, our conclusion is given in Section V.

## II. PROBLEM FORMULATION

We introduce the notations used in this paper and formally define the RDL routing problem for FC packages. Fig. 2(a)

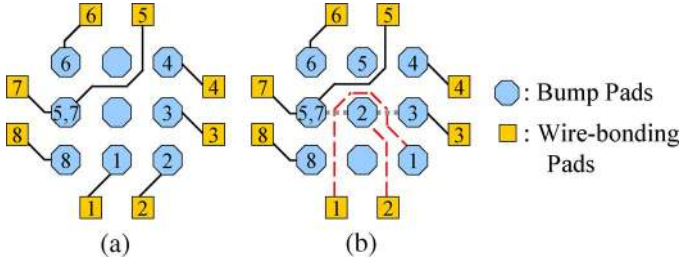


Fig. 3. (a) Monotonic routing. (b) Nonmonotonic routing.

shows the modeling of the routing structure of the FC package. Let  $P$  be the set of wire-bonding pads, and let  $B$  be the set of bump pads. For practical applications, the number of bump pads is larger than or equal to that of wire-bonding pads, i.e.,  $|B| \geq |P|$ , and each bump pad can be routed to more than one wire-bonding pad. Let  $R_b = \{r_1^b, r_2^b, \dots, r_f^b\}$  be a set of  $f$  bump pad rings in the center of the package, and let  $R_p = \{r_1^p, r_2^p, \dots, r_g^p\}$  be a set of  $g$  wire-bonding pad rings at the boundary of the package. Each bump pad ring  $r_i^b$  consists of a set of  $q$  bump pads  $\{b_1^i, b_2^i, \dots, b_q^i\}$ , and each wire-bonding pad ring  $r_j^p$  consists of  $l$  wire-bonding pads  $\{p_1^j, p_2^j, \dots, p_l^j\}$ . Let  $N$  be the set of nets (could be two-pin or multipin nets) for routing. Each two-pin (multipin) net  $n$  in  $N$  is defined by a wire-bonding pad (a set of wire-bonding pads) and a bump pad that should be connected. Since the RDL routing for current technology is typically on a single layer, it does not allow *wire crossings*, for which two wires intersect each other in the routing layer. As shown in Fig. 2(a), based on the four sides of the FC package, we partition the wire-bonding pads into four parts: *Top*, *Right*, *Bottom*, and *Left* sides. As shown in Fig. 2(b), a ring area is an area between two adjacent pad rings. A segment denotes a part of a net that connects pads or *passing points*. Nets pass through these passing points to connect wire-bonding pads and bump pads. The spacing rules for all nets are the same. A cut line is a line in the middle of two adjacent segments.

Let  $U$  be a set of intervals. We define an *interval*  $u \in U$  to be the segment between two adjacent bump pads or the segment between two adjacent wire-bonding pads in the same ring  $r_j^p$ . Given an FC routing instance, there are two types of routing—the *monotonic routing* and the *nonmonotonic routing*. Informally, a monotonic routing is a route with no U-turn path. As shown in Fig. 3(a), nets  $n_1$  (connection between wire-bonding pad 1 and bump pad 1) and  $n_2$  are monotonic routes. If we reassign the bump pads 1 and 2 as shown in Fig. 3(b), the routing of  $n_1$  becomes nonmonotonic routing. Since the nonmonotonic routing consumes more routing resource, it may result in lower routing completion.

Furthermore, the *signal skew*, i.e., the difference in wire-length between two nets, should also be considered for the routing in a high-performance FC design.

We formally define the addressed routing problem as follows.

**Problem 1:** The single-layer preassignment routing problem in the FC design is to connect a set  $P$  of wire-bonding pads and a set  $B$  of bump pads according to a predefined netlist with wire width and signal skew constraints, so that no wire crosses each other, no signal skew constraint is violated, and the total wirelength is minimized under the 100% routability guarantee.

### III. RDL ROUTING ALGORITHM

In this section, we present our routing algorithm. We first give an overview of our algorithm.

#### A. Algorithm Overview

In the routing flow, as shown in Fig. 4, our algorithm consists of two stages: 1) global routing based on *ILP* and 2) detailed routing based on passing-point assignment, net-ordering determination, and X-based gridless routing.

In the first stage of global routing, we construct a routing network  $G$  to formulate the routing of the wire-bonding pads to the bump pads (two-pin and multipin nets) as *ILP*. Since we have only one layer for routing, the *ILP* must avoid creating any wire crossings. We also formulate the wire width and signal skew constraints into the *ILP*. Since the *ILP* is NP-complete [9], it is computationally expensive. We then provide three *ILP* reduction techniques to reduce the numbers of variables and constraints. Furthermore, the nonmonotonic routes are also considered. Finally, an *ILP* solver is used to solve the *ILP* and find the routes from wire-bonding pads to bump pads. The routes give the global-routing paths of the nets.

In the second stage of detailed routing, we use passing-point assignment, net-ordering determination, and X-based gridless routing to determine the detailed routes. A *passing point* is the point for a net to pass through an interval. First, we find the passing points for all nets passing through the same interval. For all nets that pass through the same interval, we distribute these passing points according to their wire widths. We then apply net-ordering determination to each ring area to route all nets. Finally, we use an X-based gridless router to route the nets.

#### B. Global Routing

Here, we first show routing network  $G$  and the basic *ILP* formulation for routing two-pin and multipin nets. Then, we detail the three *ILP* reduction techniques for reducing the number of variables and *ILP* constraints. Finally, we discuss how to handle the routing among four sides and route nonmonotonic nets.

1) *Basic ILP Formulation:* First, we describe how to construct routing network  $G$  to perform the *concurrent routing* for the *Bottom* side. The other three sides can be processed similarly. As shown in Fig. 5(a), we define  $D = \{d_1, d_2, \dots, d_h\}$  to be a set of  $h$  *ILP nodes*. Each *ILP node* represents a candidate node of a net to pass through an interval  $(b_z^i, b_{z+1}^i)$   $((b_z^i, b_{z+1}^i))$  between two adjacent bump pads or an interval  $(p_y^j, p_{y+1}^j)$  in a wire-bonding pad ring. Let  $M$  be a set of tiles. Each *tile*  $m \in M$  represents a rectangle  $(p_y^j, p_{y+1}^j, p_{y'+1}^{j+1}, p_{y'+1}^{j+1})$   $((b_z^i, b_{z+1}^i, b_{z'+1}^{i+1}, b_{z'+1}^{i+1}))$  between two adjacent wire-bonding (bump) pad rings. We construct a routing network  $G = (P_B \cup D \cup B, E)$  for the *Bottom* side. Let  $E$  denote a set of edges that are the candidate segments of the global-routing paths of nets. There are four types of edges:

- 1) the directed edge from a wire-bonding pad to a bump pad;
- 2) the directed edge from a wire-bonding pad to an *ILP node*;

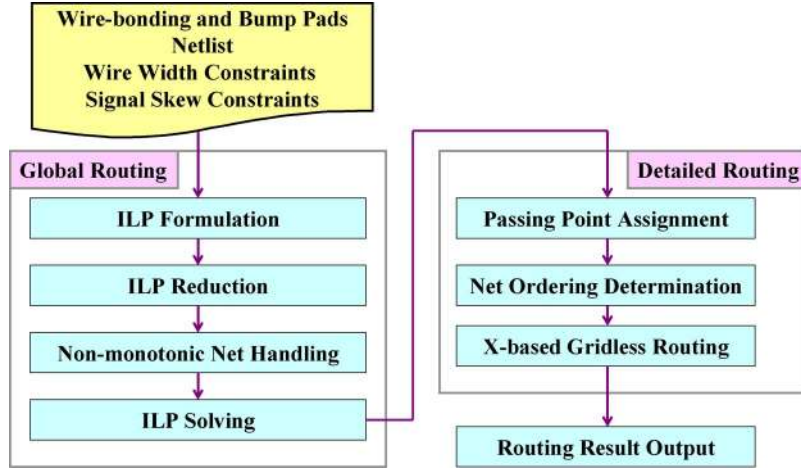


Fig. 4. RDL routing flow.

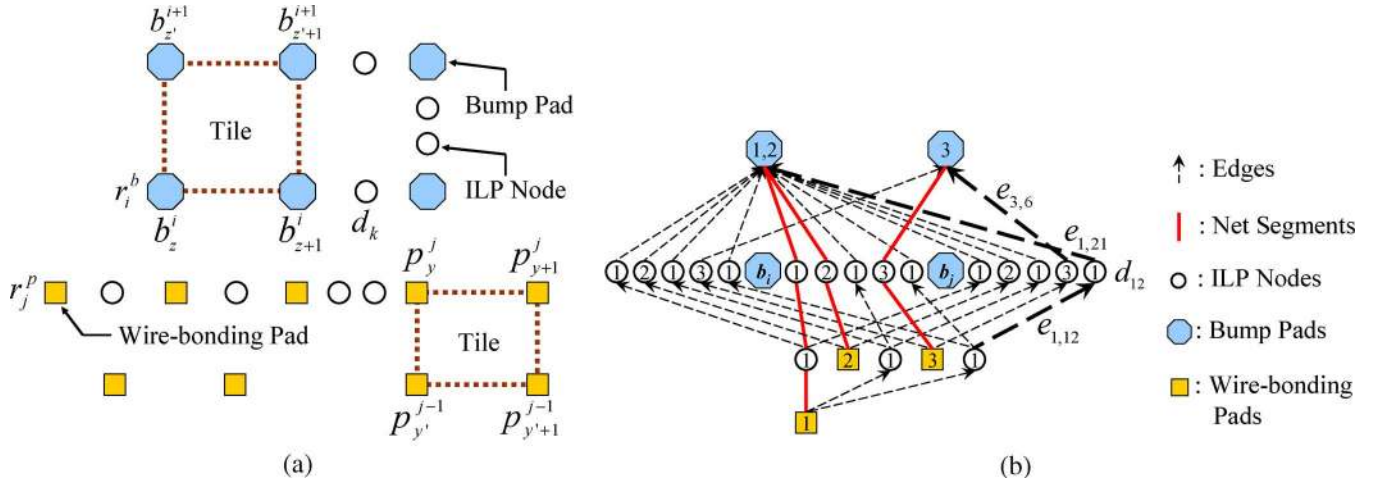


Fig. 5. (a) Intervals and tiles. (b) Routing network of the bottom side.

- 3) the directed edge from an ILP node to a bump pad;
- 4) the directed edge from an ILP node to another ILP node.

Fig. 5(b) shows an example of the routing network for the *Bottom* side. The pads and the nodes with the same number belong to the same net. Since we decompose all multipin nets into two-pin nets, nets 1 and 2 form a multipin net. We construct the routing network to contain all monotonic routing solutions to the RDL routing problem. Thus, in Fig. 5(b), we can find all global-routing paths of each net, and the solid edges denote the best solution.

The notations that are used in the ILP formulation are as follows.

- 1)  $x_{i,j}$ : 0–1 integer variable that denotes if a candidate segment  $j$  is chosen in the global-routing path of net  $n_i$ .  $x_{i,j} = 1$  if segment  $j$  is chosen;  $x_{i,j} = 0$  otherwise.
- 2)  $e_{i,j}$ : edge that denotes a candidate segment  $j$  of the global-routing path of  $n_i$ .
- 3)  $L(e_{i,j})$ : function that denotes the length of  $e_{i,j}$ .
- 4)  $W(e_{i,j})$ : function that denotes the wire width of net  $n_i$ .
- 5)  $C(e_{i,j}, e_{p,q})$ : function that denotes the crossing between  $e_{i,j}$  and  $e_{p,q}$ . If  $e_{i,j}$  crosses  $e_{p,q}$ ,  $C(e_{i,j}, e_{p,q}) = 1$ ; otherwise,  $C(e_{i,j}, e_{p,q}) = 0$ .

- 6)  $P_i(e_{i,j})$ : function that denotes the connection of  $e_{i,j}$  and wire-bonding pad  $p_i \in P$ . If  $e_{i,j}$  connects  $p_i$ ,  $P_i(e_{i,j}) = 1$ ; otherwise,  $P_i(e_{i,j}) = 0$ .
- 7)  $D_k^{\text{in}}(e_{i,j})$ : function that denotes the connection of  $e_{i,j}$  and the input side of ILP node  $d_k \in D$ . If  $e_{i,j}$  connects the input side of  $d_k$ ,  $D_k^{\text{in}}(e_{i,j}) = 1$ ; otherwise,  $D_k^{\text{in}}(e_{i,j}) = 0$ .
- 8)  $D_k^{\text{out}}(e_{i,j})$ : function that denotes the connection of  $e_{i,j}$  and the output side of ILP node  $d_k \in D$ . If  $e_{i,j}$  connects the output side of  $d_k$ ,  $D_k^{\text{out}}(e_{i,j}) = 1$ ; otherwise,  $D_k^{\text{out}}(e_{i,j}) = 0$ .
- 9)  $T_m(e_{i,j})$ : function that denotes the existence of  $e_{i,j}$  in tile  $m \in M$ . If  $e_{i,j}$  is in tile  $m$ ,  $T_m(e_{i,j}) = 1$ ; otherwise,  $T_m(e_{i,j}) = 0$ .
- 10)  $t_m$ : constant that denotes the routing resource of tile  $m \in M$ .
- 11)  $H_u(e_{i,j})$ : function that denotes the existence of  $e_{i,j}$  in interval  $u \in U$ . If  $e_{i,j}$  is in interval  $u$ ,  $H_u(e_{i,j}) = 1$ ; otherwise,  $H_u(e_{i,j}) = 0$ .
- 12)  $h_u$ : constant that denotes the routing resource of interval  $u \in U$ .
- 13)  $s_{i,p}$ : constant that denotes the maximum allowance of the signal skew between net  $i$  and net  $p$ . Each  $s_{i,p}$  is in constraints  $F$ .

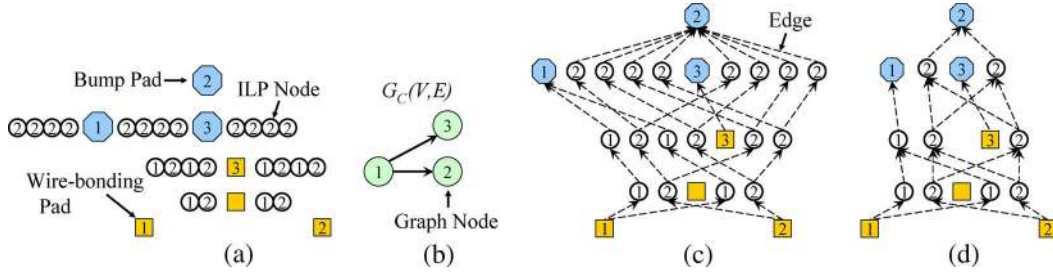


Fig. 6. (a) Routing network. (b) Constraint graph. (c) Reduced routing network after constraint-graph-based pruning. (d) Reduced routing network after ILP node merging.

Therefore, the RDL routing problem can be formulated as follows:

$$\min \sum_{e_{i,j} \in E} L(e_{i,j})x_{i,j}$$

subject to

$$C(e_{i,j}, e_{p,q})(x_{i,j} + x_{p,q}) \leq 1, \quad \forall e_{i,j}, e_{p,q} \in E \quad (1)$$

$$\sum_{e_{i,j} \in E} W(e_{i,j})T_m(e_{i,j})x_{i,j} \leq t_m, \quad \forall m \in M \quad (2)$$

$$\sum_{e_{i,j} \in E} W(e_{i,j})H_u(e_{i,j})x_{i,j} \leq h_u, \quad \forall u \in U \quad (3)$$

$$\left| \sum_{j \in n_i} L(e_{i,j})x_{i,j} - \sum_{q \in n_p} L(e_{p,q})x_{p,q} \right| \leq s_{i,p}, \quad \forall s_{i,p} \in F \quad (4)$$

$$\sum_{e_{i,j} \in E} P_i(e_{i,j})x_{i,j} = 1, \quad \forall p_i \in P \quad (5)$$

$$\sum_{e_{i,j} \in E} D_k^{out}(e_{i,j})x_{i,j} = \sum_{e_{i,q} \in E} D_k^{in}(e_{i,q})x_{i,q}, \quad \forall d_k \in D. \quad (6)$$

The objective function is to minimize the total wirelength under the 100% routability guarantee. Constraint (1) avoids the crossing: If two edges cross each other, at most one can exist. As the example shown in Fig. 5(b), since  $e_{1,21}$  and  $e_{3,6}$  cross each other, we have  $C(e_{1,21}, e_{3,6}) = 1$ . To pick at most one edge between  $e_{1,21}$  and  $e_{3,6}$ , we set  $x_{1,21} + x_{3,6} \leq 1$ . Constraint (2) is used to avoid the congestion overflow of a tile since there may be too many edges passing through the tile formed by four bump pads. We also must avoid the congestion overflow of an interval between two pads; therefore, we have constraint (3) for the edges passing through the same interval. Since the spacing rules for all nets are the same, we can use the wire width to simplify constraints (2) and (3). Note that the congestion avoidance gives the reason why we can handle variable wire widths and guarantee 100% routability after global routing. Now, we consider the signal-skew constraint. Constraint (4) formulates the signal-skew constraint between two nets. The difference in the wirelength between the two nets must be smaller than the skew constraint. Moreover, since we want to guarantee 100% routability, constraint (5) guarantees that at least one edge of the wire-bonding pad  $p_i$  of net  $n_i$  be chosen. Furthermore, as shown in Fig. 5(b), the flow summation of the output side of ILP node  $d_{12}$  must equal that of the input

side of  $d_{12}$ . For example, output flow  $e_{1,21} = 1$  if and only if input flow  $e_{1,12} = 1$ . Hence, we have constraint (6) for the flow conservation.

We have completed the basic formulation. However, this naive formulation may result in very long running time. As illustrated in Fig. 5(b), we have so many edges (variables) and crossings (constraints) for this simple problem with only three nets. For modern FC designs, there may be hundreds of nets, which may result in millions of variables and constraints. It is, thus, desirable to reduce the problem size (i.e., the numbers of variables and constraints).

2) *Optimality-Preserving ILP Reductions*: Now, we present three ILP reduction techniques to reduce the size of routing network  $G$  and, thus, the numbers of variables and constraints in the ILP. All monotonic RDL routes of a net are possible monotonic routes of the net. We define a *feasible* monotonic RDL route of a net to be the route that will not make other nets routed nonmonotonically. The three reduction techniques keep all feasible monotonic RDL routes for each net. In other words, we only prune redundant monotonic RDL routes, which will not affect the resulting routability. To maintain the solution optimality of the ILP, we shall delete only the redundant solutions.

- *Constraint-graph-based pruning*. We first consider the constraint graph for routing associated with one side of the FC for easier presentation. We will consider the net interaction handling between two sides in Section III-B3.

Fig. 6(a) shows all the ILP nodes of the routing network. According to the netlist, we can find relation among pads to avoid crossings. For example, wire-bonding pads 1 and 2 belong to the same wire-bonding pad ring. To get a monotonic route, net 1 must be routed at the left side of net 2. Similar to [15], as shown in Fig. 6(b), we can create a constraint graph  $G_C(V, E)$  to record this relation. Each vertex  $v_i \in V$  corresponds to a net  $n_i \in N$ , and each edge  $e \in E$  denotes the relative position constraint of two nets. If  $n_i$  must be routed at the left side of  $n_j$ , we construct an edge from  $v_i$  to  $v_j$ . For example, an edge is constructed from  $v_1$  to  $v_2$ . Also, we need to construct an edge from  $v_1$  to  $v_3$  because bump pads 1 and 3 are placed on the same horizontal line, and bump pad 1 is at the left side of bump pad 3. We only need to construct edges among the adjacent nets whose pins are placed in the same wire-bonding pad ring or on the same horizontal line defined by bump pads since we can find the relation between two nets by searching the constraint graph. Therefore, in this example,  $n_2$  can be routed either at the left side or the

```

Algorithm: ILP_Node_Merging( $D$ )
 $D$ : set of ILP nodes in an interval;
1 begin
2   Let  $s(i, j)$  be the net order between  $d_i$  and  $d_j$ ;
3    $l \leftarrow 0$ ;
4    $r \leftarrow \lfloor |D|/2 \rfloor$ ;
5   while  $l < r$ 
6     for  $i \leftarrow 1$  to  $|D|$ 
7       if  $(s(i, i+l) \equiv s(i+l+1, i+2l+1))$ 
8         for  $k \leftarrow i+l+1$  to  $i+2l+1$ 
9           Reconnect the edges of  $d_k$  to  $d_{k-l-1}$ ;
10          Remove the ILP nodes between  $d_{i+l+1}$ 
11            and  $d_{i+2l+1}$  from  $D$ ;
12           $r \leftarrow \lfloor |D|/2 \rfloor$ ;
13        else
14           $i \leftarrow i+1$ ;
15         $l \leftarrow l+1$ ;
16 end
    
```

Fig. 7. Algorithm for ILP node merging.

right side of  $n_3$  since there is no constraint edge between  $v_2$  and  $v_3$ . With the constraint graph, the reduced routing network of the instance in Fig. 6(a) is given in Fig. 6(c). If there exist cycles in  $G_C(V, E)$ , however, we may not find a sequence for monotonic routing. We will show how to handle nonmonotonic routing in Section III-B4.

- **ILP node merging.** We can merge some ILP nodes without losing the solution optimality. As an example shown in Fig. 6(c), at both sides of wire-bonding pad 3, there are two kinds of repetitions of the ILP nodes, such as  $\langle 2, 2 \rangle$  and  $\langle 1, 2, 1, 2 \rangle$ . We can identify all repetitions and merge them into a nonrepeated order, such as  $\langle 2 \rangle$  and  $\langle 1, 2 \rangle$ , respectively. Fig. 6(d) illustrates the routing network after merging the ILP nodes of Fig. 6(c). It is clear that the numbers of the ILP nodes and edges are reduced significantly. Note that the reduction is often very significant due to the propagation of the ILP nodes. The ILP nodes are propagated from the outer ring to the inner one, and so is the order of nets. It shows that this reduction, such as  $\langle n_i, n_j, n_i, n_j \rangle$  into  $\langle n_i, n_j \rangle$ , will not destroy the optimality. For example, if net  $n_i$  must be routed at the left side of net  $n_j$  in the outer ring, this order must be kept in the inner one for the optimal wirelength. Fig. 7 summarizes the ILP node merging algorithm; line 4 shows the maximum length of a repetition; lines 5 and 6 search each length of repetitions; line 7 finds the repetitions among ILP nodes; lines 8–11 merge the repeated ILP nodes and remove the repeated ILP nodes.
- **ILP edge bounding.** If the outgoing (incoming) edges of an ILP node between two wire-bonding (bump) pads cross all edges of the left/right pad, they are bounded by the edges of the left/right pad. The reason is that we must choose at least one edge of each wire-bonding (assigned bump) pad. As shown in Fig. 8(a), the dotted (blue) edges are deleted because they run out of bound of the solid (red) edges of wire-bonding pad 2 and bump pad 2. The routing network in Fig. 8(b) shows the ILP edge reduction result. Consequently, the number of variables and constraints can be further reduced.

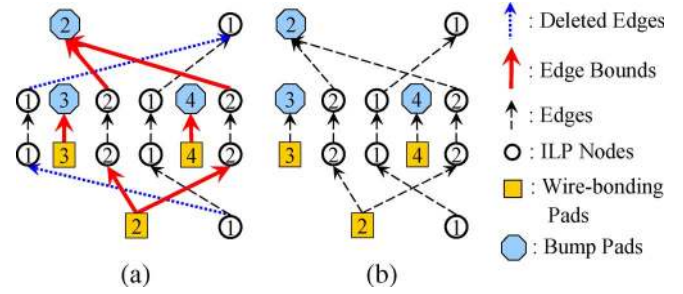


Fig. 8. (a) Routing network. (b) Reduced routing network after ILP edge bounding.

We have the following theorem for the complexity reduction by applying the above techniques.

**Theorem 1:** Given set  $P$  of wire-bonding pads, set  $B$  of bump pads, and set  $N$  of nets, the number of edges of the routing network  $G = (V, E)$  can be reduced from  $O(|N|a^a)$  to  $O(|N|a^3)$ , where  $a = \sqrt{|B|}$ , with the constraint-graph-based pruning and the ILP node merging. Consequently, the number of ILP variables can be reduced from  $O(|N|a^a)$  to  $O(|N|a^3)$ , and the number of ILP constraints can be reduced from  $O(|N|^2|B|^a)$  to  $O(|N|^2|B|^3)$ .

**Proof:** The complexity of the numbers of variables and constraints can be further reduced by constraint-graph-based pruning and ILP node merging. Given set  $P$  of wire-bonding pads and set  $B$  of bump pads for netlist  $N$  with set  $F$  of ILP constraints, we construct a routing network  $G = (V, E)$  and formulate the routing problem into the ILP. The number of ILP variables is  $O(|E|)$ . The number of ILP constraints is  $O(|E|^2)$  because it is dominated by constraint (1). Without loss of generality, we make the tile number of a row equal to that of a column. Therefore, a net can pass through at most  $a - 1$  intervals, where  $a = \sqrt{|B|}$ , i.e., the number of bump pad rings. Without the reduction,  $|E|$  is equal to  $O(|N|a^a)$  since there are at most  $a - 1$  bump pad rings passed through by a net in each side, and, thus, the edge number for a net is  $(a + 1)^1 + (a + 1)^2 + \dots + (a + 1)^{a-1} = O(a^a)$ . Here,  $a + 1$  is the maximum number of intervals in a bump pad ring. We represent the routing network after the reduction  $G' = (V', E')$ . We shall discuss only the worst-case scenario for the constraint-graph-based pruning and the ILP node merging. For the constraint-graph-based pruning, no edge and node can be reduced by the pruning when the constraint graph contains no edge. However, there may exist many repetitions of ILP nodes in each interval, and, thus, the ILP node merging can reduce  $|E|$  to  $|E'| = O(|N|a^3)$ . The reason is that each net has only one ILP node in each interval among  $a - 1$  bump pad rings, and, thus, the edge number for a net is  $(a + 1)^1 + (a + 1)^2 + \dots + (a + 1)^2 = (a + 1) + (a - 2) \times (a + 1)^2 = O(a^3)$ . For the worst-case scenario of the ILP node merging, no edge and node can be reduced by the merging when the constraint graph induces only one net order. However, the constraint-graph-based pruning changes  $|E|$  to  $|E'| = O(|N|a)$  because each net can find only one path. For other cases, the constraint-graph-based pruning and the ILP node merging can further reduce the numbers of nodes and edges in the routing network; this is because when we increase the number of edges in the constraint graph, some ILP nodes can be further pruned. However, the

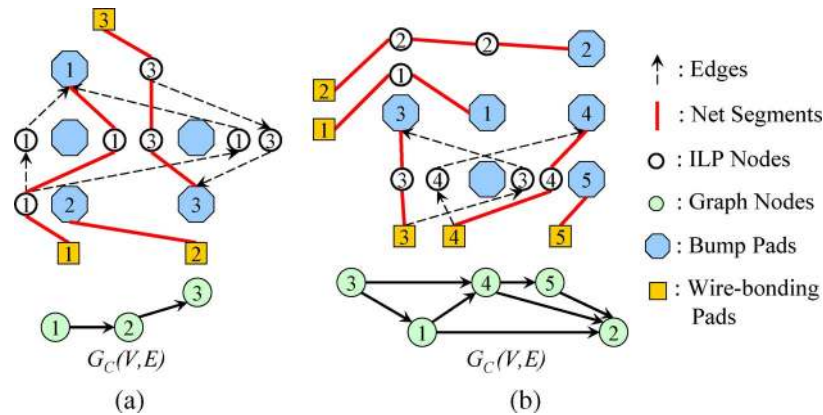


Fig. 9. (a) Routing network of parallel sides. (b) Routing network of orthogonal sides.

remaining repetitions of ILP nodes can still be merged. For example, if  $n_k$  is pruned from  $\langle n_i, n_j, n_k, n_i, n_j, n_k \rangle$ , the remaining repetition  $\langle n_i, n_j, n_i, n_j \rangle$  can be merged into  $\langle n_i, n_j \rangle$ . Hence, the reduced number of variables (edges) is  $O(|N|a^3)$ . Consequently, the number of constraints can be reduced to  $O(|N|^2a^6) = O(|N|^2|B|^3)$ . ■

Note that the reduction is very significant. With the reduction, the problem size can be reduced from exponential to polynomial.

3) *Constraint-Graph-Based Pruning With Net Interactions Between Two Sides*: We continue our discussion in the preceding section on the constraint-graph-based pruning for ILP reduction by considering the net interactions between two sides. After independently constructing routing network  $G$  for each side, the relation among different sides can be modeled as follows.

- *Parallel sides*. The parallel sides refer to the top and bottom sides or the left and right sides. As shown in Fig. 9(a), we can use the same method to construct constraint graph  $G_C$  for the top side and the bottom side. For example, since bump pad 2 is placed at the left side of bump pad 3 in the same horizontal line, an edge is constructed from vertex 2 to vertex 3, and then net 2 must be routed at the left side of net 3. Net 1 also must be routed at the left side of net 3 by searching the constraint graph. This order must be kept for routing from the bottom side toward the top side and *vice versa*. Thus, we can simultaneously route the nets of the parallel sides.
- *Orthogonal sides*. The orthogonal sides refer to the two sides with a common corner of the FC. Since the relation of the orthogonal sides is 2-D, we have to modify the construction method of  $G_C$ . As shown in Fig. 9(b), for the left and bottom sides, we can construct edges for the wire-bonding pads as before. However, for each net of the left side, we have to additionally consider the relation of the bump pads in the horizontal direction, such as bump pads 3, 1, and 4. For example, since bump pad 3 is at the left side of bump pad 1, an edge is constructed from vertex 3 to vertex 1. This additional edge denotes that net 3 must be routed at the left side and the bottom side of net 1. For each net of the bottom side, the relation of the bump pads in the vertical direction is also considered.

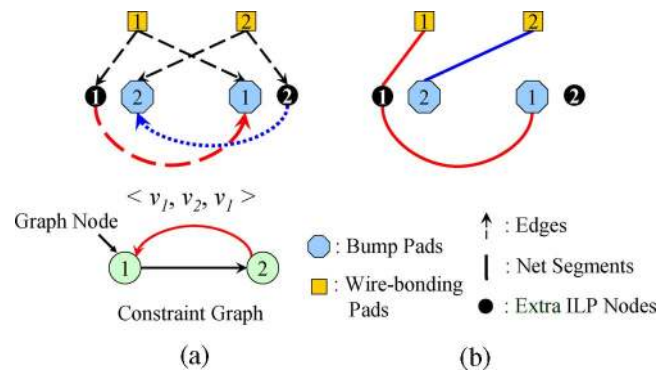


Fig. 10. (a) Simple nonmonotonic routing network. (b) Simple nonmonotonic routing result.

Bump pad 2 is at the top side of bump pad 4, so an edge is constructed from vertex 4 to vertex 2. Now, we can also simultaneously route the nets of the orthogonal sides. Note that we do not construct the edge of vertices 5 and 4 since they belong to the same side, and, thus, their relation is already constructed as discussed in the preceding section.

We refer to the ILP after applying the aforementioned three network reduction techniques as the *reduced ILP*. As discussed above, the three reduction techniques will prune only redundant solutions with nonmonotonic routes. We, thus, have the following theorem for the solution optimality (i.e., minimum wirelength) of the reduced ILP with monotonic RDL routing during global routing.

*Theorem 2*: During global routing, the reduced ILP can keep all feasible monotonic RDL routes of each net.

*Proof*: For our basic ILP formulation, it enumerates all possible RDL routes for each net. From the above discussions, we prune only the redundant solutions with nonmonotonic routes by the constraint-graph-based pruning in Sections III-B2 and B3. Then, we apply the ILP node merging to merge the feasible monotonic routes of each net in Section III-B2, which still keeps all feasible monotonic solutions. Finally, the ILP edge bounding in Section III-B2 is performed to further prune redundant solutions, keeping all feasible monotonic solutions for each net. Therefore, the reduced ILP can keep all feasible solutions with monotonic RDL routes during global routing. ■

Applying an ILP solver, we can find an optimal solution with the minimum wirelength if such a solution exists since

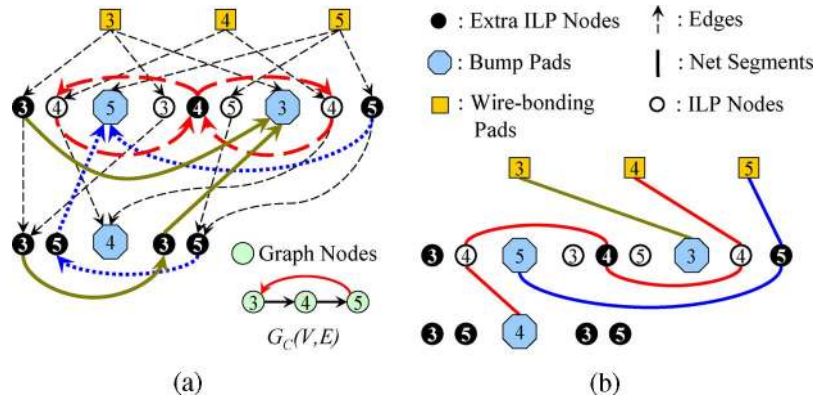


Fig. 11. (a) Complicated nonmonotonic routing network. (b) Complicated nonmonotonic routing result.

all feasible monotonic RDL routes of each net are maintained based on this theorem.

4) *Nonmonotonic Net Handling:* Here, we show two major types of nonmonotonic assignments and their ILP formulations. Other types of nonmonotonic assignments can be handled similarly. Our underlying idea is to make a nonmonotonic net monotonically routable by dividing the net into several monotonic wires. Fig. 10 gives a simple example of two nets. For the top side shown in Fig. 10(a), there exists a cycle in the constraint graph. Therefore, nonmonotonic nets may exist, and we shall search all cycles for each net. For example, the cycle of net 1 ( $n_1$ ) is  $\langle v_1, v_2, v_1 \rangle$ . We then add extra (black) ILP nodes for each net in the cycle to make the routing monotonic. In this example, the wire-bonding pads for nonmonotonic routes are in the order net 1 and net 2. Therefore, we add the extra ILP node of bump pad 2 at the right side of bump pad 1 to route over the south of bump pad 1. Then, we add the other extra ILP node of bump pad 1 at the left side of bump pad 2 to route over the south of bump pad 2. While adding these extra ILP nodes, we still follow the net order of the monotonic routing, i.e.,  $\langle n_1, n_2 \rangle$ . Hence, we can also make the ILP for  $n_1$  and  $n_2$  a monotonic one. See Fig. 10(b) for the nonmonotonic routing result.

Fig. 11 illustrates an example of three nets. For the top side shown in Fig. 11(a), there also exists a cycle in the constraint graph. We then add extra (black) ILP nodes for each net in the cycle. In this example, the wire-bonding pads for nonmonotonic routes are in the order 3, 4, and 5. We add the extra ILP node of bump pad 3 at the left side of bump pad 5 to route over the south of bump pad 5. Then, we add the other extra ILP nodes of bump pads 3 and 5 at the bottom side to route over the south of bump pad 4. Furthermore, the extra ILP node of bump pad 4 at the top side is inserted between bump pads 3 and 5 to perform nonmonotonic routing between bump pads 3 and 5, as shown in Fig. 11(b). Recall that the net order of the monotonic routing, i.e.,  $\langle n_3, n_4, n_5 \rangle$ , has to be followed while adding the ILP nodes and the extra ILP nodes. Thus, in Fig. 11(a), the white ILP nodes are inserted among those extra (black) ILP nodes according to this net order. See Fig. 11(b) for the nonmonotonic routing result.

Note that the solutions to our ILP formulation with nonmonotonic RDL routing should be near optimal since only the nets in the cycles of a constraint graph are allowed to be routed nonmonotonically.

Our ILP formulation guarantees no design-rule violations or wire crossings. Therefore, after global routing, all global-routing paths are routable. Based on the above discussions, we have the following theorem.

*Theorem 3:* If there exists a feasible global-routing solution computed by the ILP, the proposed algorithm can guarantee 100% detailed-routing completion.

*Proof:* In our global-routing model, the ILP formulation is optimal for monotonic routing and suboptimal for nonmonotonic routing. Since we consider the routing resource in the global-routing stage and will never route nets to exceed the capacity of an interval or a tile, it will never violate the design rules. Furthermore, since we avoid edge crossings in the ILP formulation, the final routing solution contains no wire crossings. After solving the ILP, all global-routing paths are routable in the detailed-routing stage. ■

5) *Comparison With the Order Graph:* Here, we compare the accuracy between our constraint graph and the order graph presented in [15]. Fig. 12 shows two examples. For the first example in Fig. 12(a) and (b), the order graph does not have any edge from vertex 2 to vertex 1. Therefore, it may not complete the routing by using the order graph [15]. In our constraint graph, however, since the bump pad at the right side of bump pad 2 is not assigned, we can temporarily assign net 2 to the empty bump pad and then construct this edge. By doing so, we can prune more ILP variables and, thus, the ILP constraints. In Fig. 12(c), according to the order graph, we cannot complete the routing because there exist conflicts among the three nets. As shown in Fig. 12(d), therefore, we generate the cyclic constraint graph. Then, we insert ILP nodes for nonmonotonic routing. Since our nonmonotonic routing network contains monotonic routing solutions, we can still get the monotonic routing result by using the ILP. Therefore, it is clear that our constraint graph captures the relation of nets more accurately than the order graph.

### C. Detailed Routing

The objective in detailed routing is to accomplish the routing and minimize the number of wire bends after passing-point assignment and net ordering determination. Therefore, we use a two-phase technique, as shown in Fig. 13, to perform 45° gridless detailed routing with passing points and net order. In the first phase, a method with Hanan grids [5] is used to

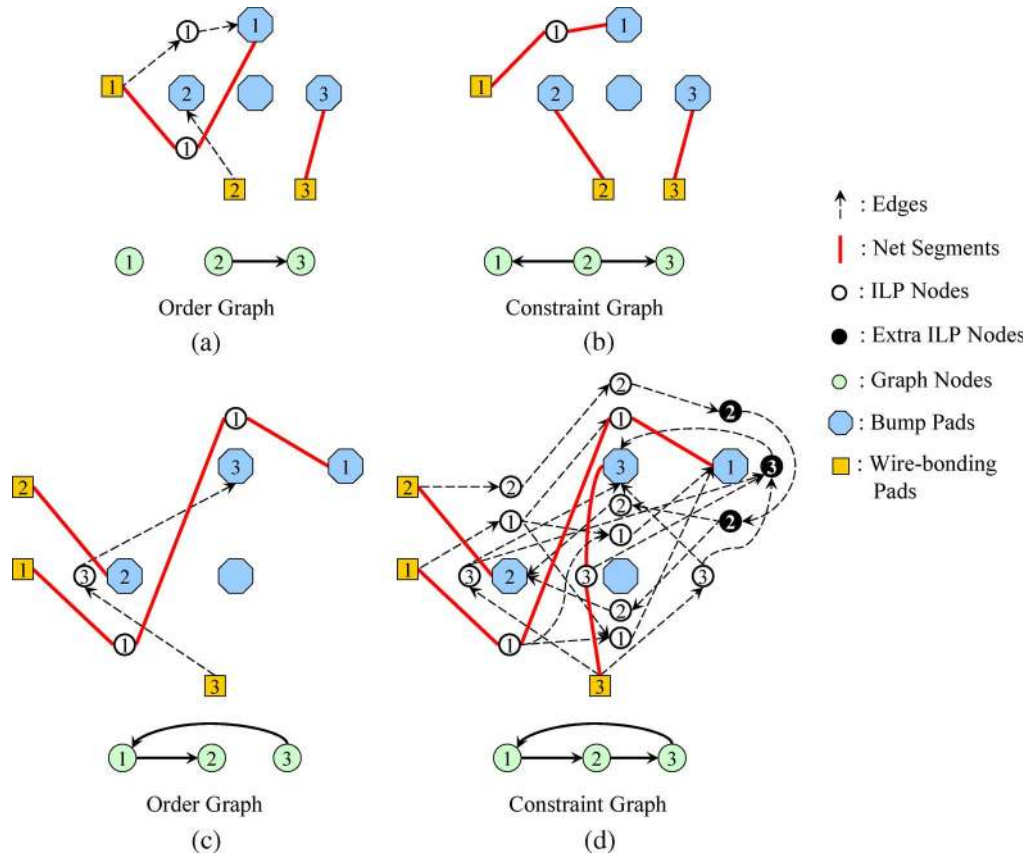


Fig. 12. (a) Wrong routing result (top) by using the order graph (bottom). (b) Correct routing result by using the constraint graph. (c) No feasible monotonic routing result by using the order graph. (d) Monotonic routing result by using the cyclic constraint graph.

```

Algorithm: Two Phase Detailed Routing( $S$ )
 $S$ : segments of nets;
1 begin
2   Choose connected cut lines;
//Phase I
3   for each ring area  $r_i$ 
4      $A_s \leftarrow$  Order array of all segments in  $r_i$  clockwise;
5     for  $j \leftarrow 0$  to  $A_s.size() - 1$ 
6        $G_j \leftarrow$  Construct Hanan grids for  $A_s[j]$ ;
7        $p \leftarrow DFS(G_j)$ ; //  $p$  is a path.
//Phase II
8    $A_n \leftarrow$  Order array of all nets counterclockwise;
9   for  $k \leftarrow 0$  to  $A_n.size() - 1$ 
10     $G_k \leftarrow$  Construct Hanan grids for  $A_n[k]$ ;
11     $P \leftarrow BFS(G_k)$ ; //  $P$  contains paths.
12     $\alpha \leftarrow (k - 1) \bmod A_n.size()$ ;
13    Record  $p \in P$  which is the closest to  $A_n[\alpha]$ ;
14   return all paths of nets;
15 end
    
```

Fig. 13. Two-phase detailed routing algorithm.

accomplish detailed routing. In the second phase, bend minimization is performed under the 100% routability guarantee.

1) *Passing-Point Assignment*: After global routing, the global-routing paths are free of wire crossings. To utilize the result of our global routing, we use a method called passing-point assignment to distribute nets that pass through the same interval, according to their wire widths and further signal-skew constraints. Passing points are transformed from ILP nodes. For example, as shown in Fig. 14, the two nets from wire-bonding

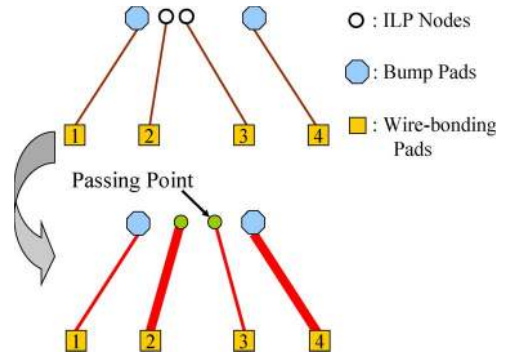


Fig. 14. Passing-point assignment.

pads 2 and 3 pass through the same interval on two ILP nodes. We assign two passing points according to the wire widths of the two nets.

2) *Net-Ordering Determination*: There is no wire crossing in our global-routing results. Thus, as shown in Fig. 2(b), we choose a cut line in every ring area and order all segments in the same ring area clockwise. No matter to which segments the cut line is adjacent, the net order that is produced by our method is routable. The reason is that every segment is compacted to the preceding segment during routing. Therefore, between two nets, we can connect these cut lines in every ring area and then determine the net order according to the connected cut lines, as shown in Fig. 2(a). For example, in Fig. 13, the algorithm orders all segments and nets in lines 4 and 8 according to the connected cut lines decided in line 2.

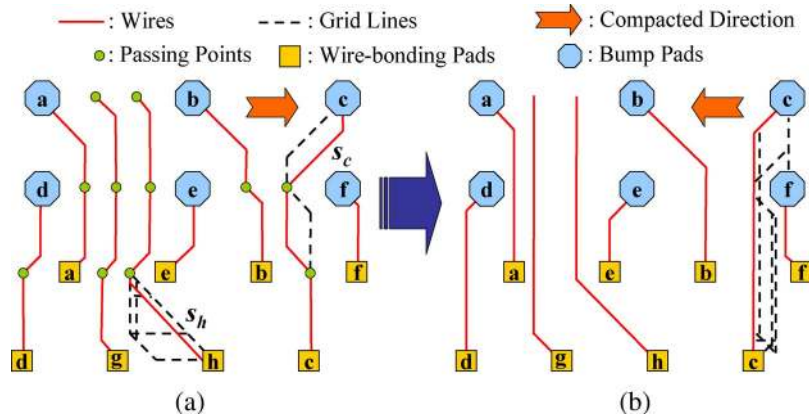


Fig. 15. (a) Routing of phase I. (b) Routing of phase II.

3) *X-Based Gridless Routing*: The X-based gridless router uses the result of net-ordering determination to complete detailed routing. It consists of two phases as follows.

- *Phase I*. In the first phase, we route all nets segment by segment and realize the compacted routing in order. Our algorithm first incrementally constructs Hanan grids for a segment. For example, in Fig. 15(a), the grids of segment  $s_h$  are more complicated than those of  $s_c$ . The reason is that using fewer grids of  $s_h$  might violate design rules. Therefore, our algorithm constructs more complex and routable grids of  $s_h$ . Then, we route each segment by depth-first search (DFS) on the grids from one terminal of a segment to the other. As shown in Fig. 15(a), dash lines are grids, and solid lines are paths of the DFS. Each path is compacted to the preceding segment according to the net order. Thus, most routing space is reserved for the next net segment, and it is easier to accomplish routing in the same ring. In Fig. 13, for instance, the first phase is performed in lines 3–7. During the DFS in line 7, a node on grids that is closer to the preceding segment or the cut line is tried first until the segment routing succeeds.

The first segment is compacted to the cut line in the same ring area. Then, we route the other net segments in the same ring area in order (clockwise). By using the result of global routing, the passing-point assignment, and the segment compaction, our algorithm can accomplish detailed routing in the first phase. However, the passing-point assignment and the segment compaction might incur redundant bends in detailed routing, as shown in Fig. 15(a). Consequently, we need to perform optimization in the second phase.

- *Phase II*. After accomplishing detailed routing in the first phase, we perform bend minimization net by net and reduce the total wirelength at the same time. Our method starts from the connected cut lines and orders all nets counterclockwise, as shown in line 8. In lines 9–13, our algorithm routes a whole net from a bump pad to a wire-bonding pad and considers obstacles, such as adjacent nets and pads, while constructing Hanan grids. It applies breadth-first search (BFS) from a bump pad on grids until reaching a wire-bonding pad. During the BFS in line 11, a node only records one fan-in node with the fewest

**Algorithm: RDL Routing( $P, B, N, F$ )**  
 $P$ : set of all wire-bonding pads;  
 $B$ : set of all bump pads;  
 $N$ : set of all 2-pin and multi-pin nets;  
 $F$ : set of all wire width and signal skew constraints;  
 $S$ : set of net segments in a ring area;  
 $S'$ : set of sorted net segments in a ring area;  
**1 begin**  
 2 Construct a routing network  $G$ ;  
 3 Formulate ILP functions of  $P$  and each assigned  
 4 bump pad  $b \in B$  simultaneously for each net  
 5  $n \in N$  with  $F$ ;  
 6 Use three optimality-preserving ILP reduction  
 7 techniques to reduce the numbers of variables  
 8 and constraints in the ILP;  
 9 Use an ILP solver to find the global-routing path  
 10 of each net  $n \in N$ ;  
 11 Find passing points in all intervals for each net  
 12  $n \in N$ ;  
 13 **for** the outermost ring  $r_i^p$  to the innermost ring  $r_j^b$   
 14 Net\_Ordering\_Determination( $S$ );  
 15 X-based\_Gridless\_Routing( $S'$ );  
 16 Reroute each net  $n \in N$  to reduce the number  
 17 of bends and total wirelength;  
 18 **return** the RDL routing result;  
 19 **end**

Fig. 16. Overview of the RDL routing algorithm.

wire bends. Once fan-in nodes have the same bends, the algorithm chooses a node whose back-traced path is closer to the preceding net, as shown in line 13. In Fig. 15(b), for example, net  $c$  is a path of the BFS with the least wire bends.

The difference among the grids in the first phase and the second phase is the passing-point consideration. In other words, a path of a net produced in the second phase will not go through the passing points of the net if these passing points increase the number of bends. Of course, this optimization is performed to each net without affecting the 100% detailed routing completion in the first phase.

#### D. Summary

We have proposed an RDL routing algorithm that consists of ILP, optimality-preserving ILP reduction techniques, and

TABLE I  
BENCHMARK CIRCUITS FOR RDL ROUTING

Circuits	#Nets (2-pin/multi-pin/non-monotonic)	#Skew pairs	#Wire widths	#Rp	#p	#Rb	#b
fc1189	513/24/9	4	2	2	513	13	676
fc1458	646/17/21	6	2	2	646	7	812
fc1795	639/6/16	6	2	2	639	17	1156
fc1813	657/4/24	6	2	2	657	17	1156
fc2624	1024/64/56	10	2	2	1024	20	1600

TABLE II  
EFFECTS OF THE REDUCTION TECHNIQUES

Ours Circuits	#Variables				#Constraints				Total wirelength ( $\mu\text{m}$ )	CPU time (min)
	RR of CG (%)	RR of NM (%)	RR of EB (%)	Total	RR of CG (%)	RR of NM (%)	RR of EB (%)	Total		
fc1189	99.9	86.3	5.6	10968	99.9	99.1	8.4	230935	500072	3.4
fc1458	99.9	72.8	8.2	17792	99.9	95.8	10.3	376090	638740	18.7
fc1795	99.9	84.5	9.1	18867	99.9	95.6	11.0	1185272	571996	35.7
fc1813	99.9	90.6	4.6	20591	99.9	99.7	9.1	999679	533947	28.1
fc2624	99.9	85.9	7.3	39788	99.9	99.3	8.5	832825	1412854	55.8
Average	99.9	84.0	7.0		99.9	97.9	9.46			

X-based gridless routing to optimize the routability, the total wirelength, and the signal skew. The whole algorithm is illustrated in Fig. 16. Lines 2–5 formulate the routing network by the ILP; lines 6–10 use three reduction techniques to find the global-routing paths; lines 11–17 show how to determine the net order and then apply the X-based gridless routing to complete the routing.

#### IV. EXPERIMENTAL RESULTS

We implemented our algorithm in the C++ programming language on a 2.6-GHz AMD Opteron Linux workstation with 6-GB memory. We used the public `lp_solve` [10] to solve the ILP. The benchmark circuits, which are listed in Table I, are real industry designs with predefined netlists. In Table I, “Circuits” denotes the names of circuits, “#Nets” denotes the number of nets, “#Skew pairs” denotes the number of matched nets with skew constraints, “#Wire widths” denotes the number of variable wire widths, “# $R_p$ ” denotes the number of wire-bonding pad rings, “# $p$ ” denotes the number of wire-bonding pads, “# $R_b$ ” denotes the number of bump pad rings, and “# $b$ ” denotes the number of bump pads.

Two experiments were performed to verify our router. In the first experiment, we explored the effects of the three reduction techniques on the problem sizes presented in Section III-B2. For this experiment, we routed the five circuits with the predefined netlists, including wire-width constraints, signal-skew constraints, and nonmonotonic nets based on our algorithm with and without the reductions. The experimental results are shown in Table II. Since the routability is all 100% for all circuits, and no skew constraints are violated, we do not list them in the table. Instead, we focus on the numbers of variables and constraints of the ILP with and without the reductions. “RR of CG (%)” denotes the reduction rate (RR) of the variables (constraints) by using the constraint-graph-based pruning (CG),

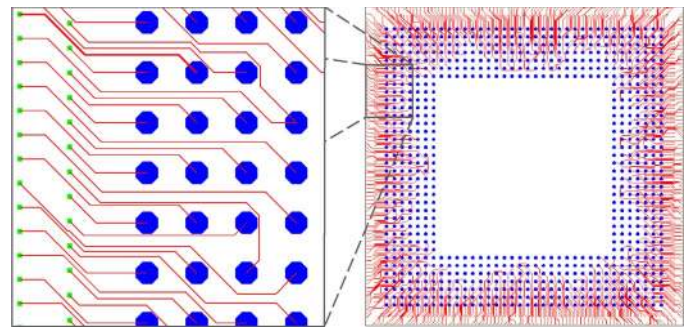


Fig. 17. RDL routing result for fc1458.

“RR of NM (%)” denotes the reduction rate over the results listed in the column “RR of CG (%)” by using the ILP node merging (NM), “RR of EB (%)” denotes the reduction rate over the results listed in the column “RR of NM (%)” by using the ILP edge bounding (EB), and “Total” gives the final resulting number of variables (constraints). As shown in the table, the constraint-graph-based pruning can prune more than 99.9% of the number of variables (constraints) in the basic ILP formulation, the ILP node merging can further reduce the number of variables (constraints) by an average of 84.0% (97.9%), and the ILP edge bounding can further reduce the number of variables (constraints) by an average of 7.0% (9.5%). The experimental results show the effectiveness of the three reduction techniques. As a result, our ILP-based routing algorithm can obtain the final routing results in reasonable CPU times due to the significant problem-size reduction. Fig. 17 shows the RDL routing result of fc1458.

In the second experiment, we verify the quality of our algorithm. Since there is no preassignment RDL routing algorithm for the FC design in the literature, we implemented the related work presented in [15] (originally for BGA global routing) for

TABLE III  
COMPARISON BETWEEN [15] AND OURS

Routers Circuits	Global wirelength ( $\mu\text{m}$ )				Routability (%)		#Skew pair violations		Total wirelength ( $\mu\text{m}$ )	CPU time (s)	
	[14]	Ours	RR (%)	Lower bound	[14]	Ours	[14]	Ours	Ours	[14]	Ours
fc1189_f	675052	572844	15.2	572844	88.3	100	4/4	0/4	565320	0.5	68.9
fc1458_f	888453	769620	13.4	769620	80.6	100	6/6	0/6	766582	1.1	295.4
fc1795_f	773403	634510	18.0	634510	85.7	100	6/6	0/6	615348	0.7	446.0
fc1813_f	882912	699531	20.8	699531	82.0	100	6/6	0/6	689579	1.1	393.2
fc2624_f	1773045	1585989	10.6	1585989	78.1	100	10/10	0/10	1581332	2.2	792.9
Comp.			15.6		82.9	100	32/32	0/32			

TABLE IV  
COMPARISON OF THE REDUCTION TECHNIQUES

Routers Circuits	#Variables			#Constraints			CPU time (min)	
	[14]	Ours	RR (%)	[14]	Ours	RR (%)	[14]	Ours
fc1189_f	81625	9563	88.3	25531243	216365	99.2	> 1440	1.1
fc1458_f	66020	16879	74.4	8880020	365712	95.9	> 1440	4.9
fc1795_f	121585	18331	84.9	26741211	1163128	95.7	> 1440	7.4
fc1813_f	220848	18735	91.5	262312781	973256	99.6	> 1440	6.5
fc2624_f	287075	33247	88.4	133174225	811463	99.4	> 1440	13.2
Average			85.5			98.0		

the comparative study because the BGA global router is also for the single-layer routing structure. However, detailed routing is not considered in [15]. Hence, we can only use [15] as a global router to compare with ours. Because [15] can only handle monotonic two-pin routes with the uniform wire width, we used the free-assignment RDL router presented in [6] and [8] to route the benchmark circuits and then extract the connections among wire-bonding pads and bump pads. With the connections, we obtain the netlist for the preassignment RDL routing. (Therefore, we added *\_f* to the names of the benchmarks in Table I to note the difference; see Table III.) Furthermore, [6] and [8] guarantee the minimal global wirelength and 100% routability while dividing an FC into four independent sectors. Therefore, its solution is the lower bound of the global wirelength, which can be used to verify the solution optimality of each router. Also, since we know the wirelength of each net, we can define the difference in wirelength between two nets to be the maximum allowance of the signal skew of them. We randomly chose the same number of skew pairs for each circuit, as shown in Table I. To perform a fair comparison of the routability, we also used the detailed router in [6] and [8] to complete the routing for the global route generated in [15]. Note that we do not compare with the global router in [6] and [8] because it uses the network-flow algorithm in global routing and cannot handle the preassignment RDL routing problem.

The experimental results are reported in Table III. In global routing, the completion rates of the two routers ([15] and ours) are both 100%. We define the global wirelength to be the total length of nets after global routing. The routability gives the completion rate of detailed routing. The total wirelength is

the total length of nets after detailed routing. We also report the number of skew violations and the CPU time. For each circuit, we generated 100 monotonic routing patterns for the routing algorithm in [15] as the authors did and averaged its experimental results. The experimental results show that our ILP-based algorithm can achieve 100% routability and the optimal global-routing wirelength and satisfy all signal-skew constraints under reasonable CPU times. Compared with [15], our router reduces the global wirelength by 15.56%. Furthermore, [15] combined with the detailed router in [6] and [8] can achieve only 82.9% routability and fails all signal-skew constraints. These improvements also reveal that finding a good routing sequence considering the routing resource is very important and dominates the whole routing results. Note that since [6] and [8] first divide an FC into four independent sectors and then generate the netlist, the routing resource between two adjacent sectors is not utilized. As a result, the free-assignment netlists may lead to only suboptimal solutions. However, the predefined netlists in Table II can also consider the routing resource between two adjacent sectors and, thus, may lead to better solutions. Because [15] cannot complete detailed routing, we only report our total wirelengths. Furthermore, according to the lower bounds, our router can achieve the solution optimality during global routing. In addition, the assignment of each side is independent of the others in Table III, and, thus, we can separately route each side to reduce the complexity of the ILP formulation. Thus, the CPU time in Table III is shorter than that in Table II. The results show that our ILP-based RDL routing algorithm is very effective, robust, and flexible for the FC design.

We further compare the effectiveness of the reduction techniques used in [15] (the order graph, i.e., the constraint graph considering only one side of the FC) and ours. The results are listed in Table IV. As shown in the table, we can further reduce the number of variables (constraints) by 85.5% (98.0%). Consequently, with our reduction techniques, our ILP-based RDL routing consumed only reasonable CPU times; in contrast, it is not feasible for the ILP-based routing ( $> 5$  days per circuit), with the reduction technique used in [15].

## V. CONCLUSION

We have developed an RDL router for the FC package, considering signal skews, variable wire widths, nonmonotonic routes, and total wirelength minimization. Our ILP-based algorithm guarantees to find an optimal solution for the addressed problem. Experimental results have demonstrated that our router can achieve 100% routability and the optimal global-routing wirelength and satisfy all signal-skew constraints under reasonable CPU times. The ILP-based RDL routing algorithm is very effective, robust, and flexible for the FC design.

## REFERENCES

- [1] H. Cai, "Multi-pads, single layer power net routing in VLSI circuits," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 1998, pp. 183–188.
- [2] D.-S. Chen and M. Sarrafzadeh, "A wire-length minimization algorithm for single-layer layouts," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 1992, pp. 390–393.
- [3] S.-S. Chen, J.-J. Chen, S.-J. Chen, and C.-C. Tsai, "An automatic router for the pin grid array package," in *Proc. ACM/IEEE Asia South Pacific Des. Autom. Conf.*, Jan. 1999, pp. 133–136.
- [4] S.-S. Chen, J.-J. Chen, C.-C. Tsai, and S.-J. Chen, "An even wiring approach to the ball grid array package routing," in *Proc. IEEE Int. Conf. Comput. Des.*, Oct. 1999, pp. 303–306.
- [5] C. Chiang, Q. Su, and C.-S. Chiang, "Wirelength reduction by using diagonal wire," in *Proc. GLSVLSI*, 2003, pp. 104–107.
- [6] J.-W. Fang, I.-J. Lin, P.-H. Yuh, Y.-W. Chang, and J.-H. Wang, "A routing algorithm for flip-chip design," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2005, pp. 753–758.
- [7] J.-W. Fang, C.-H. Hsu, and Y.-W. Chang, "An integer linear programming based routing algorithm for flip-chip design," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2007, pp. 606–611.
- [8] J.-W. Fang, I.-J. Lin, Y.-W. Chang, and J.-H. Wang, "A network-flow based RDL routing algorithm for flip-chip design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 8, pp. 1417–1429, Aug. 2007.
- [9] M. R. Garey and D. S. Johnson, *A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.
- [10] [Online]. Available: <http://lpsolve.sourceforge.net/5.5/>
- [11] Y. Kubo and A. Takahashi, "A global routing method for 2-layer ball grid array packages," in *Proc. ACM Int. Symp. Phys. Des.*, Apr. 2005, pp. 36–43.
- [12] M. Sarrafzadeh, K.-F. Liao, and C. K. Wong, "Single-layer global routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 13, no. 1, pp. 38–47, Jan. 1994.
- [13] R. Shi and C.-K. Cheng, "Efficient escape routing for hexagonal array of high density I/Os," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jul. 2006, pp. 1003–1008.
- [14] A. Titus, B. Jaiswal, T. J. Dishongh, and A. N. Cartwright, "Innovative circuit board level routing designs for BGA packages," *IEEE Trans. Adv. Packag.*, vol. 27, no. 4, pp. 630–639, Nov. 2004.
- [15] Y. Tomioka and A. Takahashi, "Monotonic parallel and orthogonal routing for single-layer ball grid array packages," in *Proc. ACM/IEEE Asia South Pacific Des. Autom. Conf.*, Jan. 2006, pp. 642–647.
- [16] C.-C. Tsai, C.-M. Wang, and S.-J. Chen, "News: A net-even-wiring system for the routing on a multilayer PGA package," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 2, pp. 182–189, Feb. 1998.
- [17] UMC, *0.13  $\mu\text{m}$  flip-chip layout guideline*, p. 6, 2004.



**Jia-Wei Fang** (S'05) received the B.S. degree in electrical engineering in 2003 from the National Cheng Kung University, Tainan, Taiwan, and the M.S. degree in electronics engineering in 2005 from the National Taiwan University, Taipei, Taiwan, where he is currently working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering.

His current research interests include flip-chip routing and chip-package-board codesign.



**Chin-Hsiung Hsu** (S'06) received the B.S. degree in computer science and information engineering in 2005 from the National Taiwan University, Taipei, Taiwan, where he is currently working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering.

His research interests are in combinatorial optimization with applications to the VLSI design automation, large-scale global routing, and flip-chip design.



**Yao-Wen Chang** (S'94–A'96–M'99) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1988, and the M.S. and Ph.D. degrees from the University of Texas at Austin in 1993 and 1996, respectively, all in computer science.

He is a Professor in the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University. He is currently also a Visiting Professor at Waseda University, Kitakyushu, Japan. He was with the IBM T.J. Watson Research Center, Yorktown Heights, NY, in the summer of 1994. From 1996 to 2001, he was on the faculty of National Chiao Tung University, Taiwan. His current research interests lie in VLSI physical design, design for manufacturability/reliability, and design automation for biochips. He has been working closely with industry on projects in these areas. He has coedited one textbook on EDA, coauthored one book on routing, and over 140 ACM/IEEE conference/journal papers in these areas.

Dr. Chang is a winner of both the 2008 ACM ISPD Global Routing Contest and the 2006 ACM ISPD Placement Contest. He received Best Paper Awards from ICCD-95 and the 2007 and 2008 VLSI/Design CAD Symposia, and twelve Best Paper Award Nominations from DAC (four times), ICCAD (twice), ISPD (three times), ACM TODAES, ASP-DAC, and ICCD in the past eight years. He has received many awards for research performance, such as the 2007 Outstanding Research Award, the inaugural 2005 First-Class Principal Investigator Award, and the 2004 Dr. Wu Ta You Memorial Award, all from National Science Council of Taiwan, the 2004 MXIC Young Chair Professorship from the MXIC Corp, and for excellent teaching from National Taiwan University (four times) and National Chiao Tung University.

He is currently an Associate Editor of IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (TCAD) and an editor of the *Journal of Information Science and Engineering* (JISE) and the *Journal of Electrical and Computer Engineering* (JECE). He has served on the ICCAD Executive Committee, the ASPDAC Steering Committee, the ACM/SIGDA Physical Design Technical Committee, the ACM ISPD and IEEE ICFPT Organizing Committees, and the technical program committees of ASP-DAC (topic chair), DAC, DATE, FPL, GLSVLSI, ICCAD, ICCD, ICFPT (program chair), IECON (topic chair), ISPD, SOCC (topic chair), TENCON, and VLSI-DAT (topic co-chair). He is currently an independent board director of Genesys Logic, Inc, a technical consultant of RealTek Semiconductor Corp., a member of board of governors of Taiwan IC Design Society, and a member of the IEEE Circuits and Systems Society, ACM, and ACM/SIGDA.