Board- and Chip-Aware Package Wire Planning

Ren-Jie Lee, *Student Member, IEEE*, Hsin-Wu Hsu, and Hung-Ming Chen, *Member, IEEE*

***Abstract*— The slow turnaround between design, package, and system houses has been one of the primary concerns in the semiconductor business. There is a serious lag in the development time of the systems due to time-consuming interface design between the chip, package, and board. In order to enable chip–package–board codesign to speed up the design process, we propose an approach to address this issue by efficiently planning wires for board and chip design awareness, which includes the package pin-out designation and the corresponding wire planning in package and board. We model the problem as an interval intersection problem. Because of the special need in pin-out rules, an algorithm to resolve the problem is developed. We then use some optimization techniques to further improve objectives such as global wire congestion and length deviation. Our results show that a very efficient estimation can be made considering those important objectives, and package congestion can be successfully mitigated.**

***Index Terms*— Chip-package-board codesign, package congestion mitigation, package wire planning.**

I. Introduction

TODAY, large gaps are emerging between chip, package, and board designs. A large amount of resources is spent on reaching a consensus between these three interfaces. Chip–package–board codesign targets better system performance and shorter design cycles. It efficiently facilitates achieving a convergent solution. Fig. 1 shows an example of the whole platform: signals starting from I/O pads travel through many interfaces including redistribution layer (RDL) bumps, package balls, and the printed circuit board (PCB). In modern VLSI designs, more than 1000 I/O pins are usually required to communicate with each other. Because of the demand for more I/Os, ball grid array (BGA) packaging has become a major interface between the chip and PCB. Tradeoffs between system performance and cost are therefore determined by BGA pin-out designation (also called ballout).

In [1], the authors have proposed an efficient approach to automate pin-out designation for package–board codesign. Their frameworks consider signal integrity (SI), power delivery integrity (PI), and routability (RA) in pin-out block design, and achieve close-to-minimum package size while providing good signal quality. However, more requirements need to be further fulfilled in pin-out designation, in addition to the performance metrics mentioned above, to facilitate the routing works between the chip, package, and PCB. Moreover, the design on the chip side should also be accounted for, in order to reflect important design constraints that impact package wire planning.

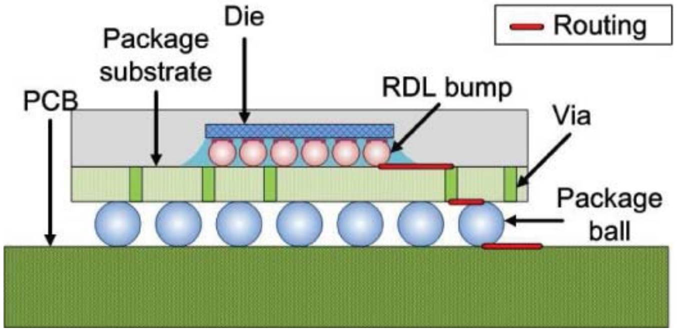


Fig. 1. Cross section of the platform: signal trace traveling through three interfaces including RDL bumps, package balls, and PCB.

A. Previous Works and Motivations

Regarding the flip-chip designs, it is generally classified into two regimes: One is called peripheral-array I/O (PIO) where bumps are placed along the chip boundary. The other is called area-array I/O (AIO) where bumps are placed in the central area of the chip [2]. Since AIO accommodates many more bumps than PIO, it is more suitable for modern VLSI designs.

For AIO flip-chip designs, some sophisticated RDL routing methods have been developed to connect the peripheral I/O pads with area-array bump pads. According to the pre-assigned order of I/O pads, Fang *et al.* applied network-flow-based [2] and integer-linear-programming-based [3] RDL routing algorithms for designing area-array ICs. Each of these two-stage techniques not only completes 100% RA but also reduces the total RDL wirelength and signal skews compared with an industrial heuristic algorithm. Consequently, in order to preserve the optimized results in RDL routing, the pin-out designation must follow the ordered I/O pin sequence while designing the package.

On the other hand, considering the PCB routing problem, it can also be divided into two categories. One is escape routing, which routes nets from the pin terminal (ball) to the component boundaries. The other one is area routing, which routes nets between component boundaries [4]. For area routing, the planar-fashioned bus routing is always preferred to control and match impedance for each high-speed signal. One approach regarding automatic bus planner for PCB was published very recently [5]. On testing a state-of-the-art industrial circuit board, their bus planner achieves 98.5% routing completion and simultaneously assigns routing layers and nets.

Manuscript received September 28, 2011; revised May 15, 2012; accepted July 21, 2012. Date of publication September 13, 2012; date of current version July 22, 2013.

R.-J. Lee is with Novatek Microelectronics Corporation, Hsinchu 300, Taiwan (e-mail: rjlee@vda.ee.nctu.edu.tw).

H.-W. Hsu is with TSMC, Hsinchu 300-78, Taiwan (e-mail: hsu.hsinwu@gmail.com).

H.-M. Chen is with the Institute of Electronics, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: hmchen@mail.nctu.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/TVLSI.2012.2212288

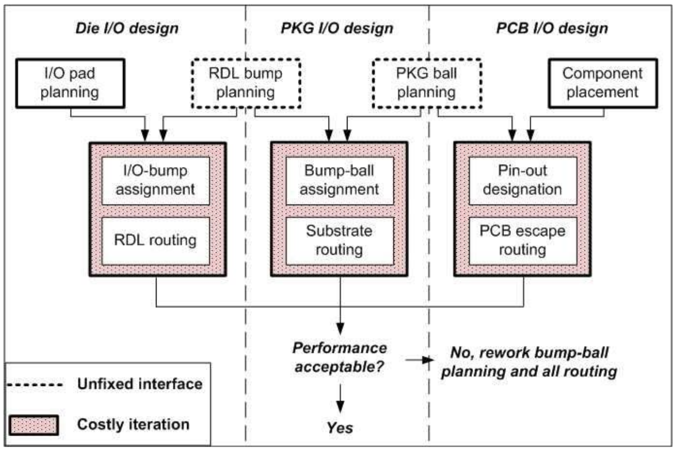


Fig. 2. Conventional flow suffering from costly rework and slow turnaround.

However, the basic requirement of this bus planner is ordered escape routing, which routes nets from balls to component boundaries with a given order. Without ordered escape routing, it is not guaranteed that the planar bus routing between components can be done [6]. To achieve ordered escape routing, the given I/O pin sequence must be carefully considered when designating the package pin-out.

B. Our Contributions

The common approach usually takes weeks to rearrange the pin-out, rework the package substrate, and lay out the PCB, as shown in Fig. 2, and each modification of the interfaces can result in costly iterations. For chip core designers, several iterations of modifying I/O pads and RDL bumps with system designers will eventually take at least one month for hundreds or even a thousand pins. We hope to have a fast estimation on the resources we can use in package and board, to skip long turnaround times and iterations between the design house, the package house, and the system house. This paper proposes a feasible pin-out designation which considers the ordered pin sequence in both the die side and the package side. These ordered pin sequences are passed to die RDL routing and PCB area routing, which are optimized by using previous schemes [2]–[6]. In other words, core designers can specify the preferred I/O pad ordering, and system designers can specify the preferred bump pin-out designation. Our method can efficiently analyze whether the preferences from both sides accommodate each other, before performing RDL routing and substrate routing. Thus the flow can be replaced by the proposed methodology, as shown in Fig. 3.

The rest of this paper is organized as follows. Section II defines the problem of wire planning for a two-layer package design and PCB escape routing considering the ordered pin sequence. Section III describes the package ballout and wire planning approach; Section IV shows the optimization for various objectives to further strengthen our methodology. Section V shows the experimental results, followed by conclusions in Section VI.

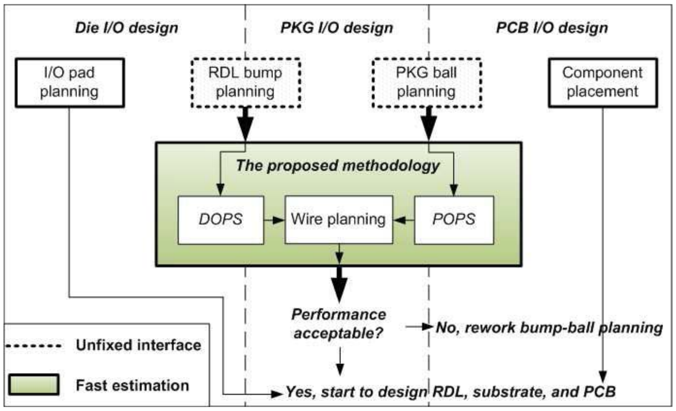


Fig. 3. Proposed flow enabling fast chip–package–board codesign respin.

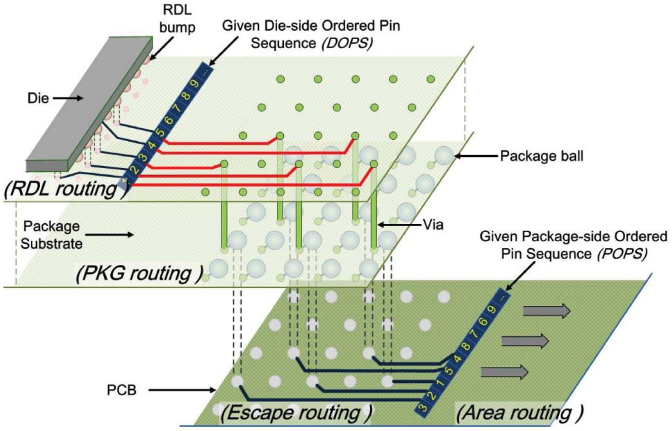


Fig. 4. Wires/traces routed in the two-layer BGA package model.

II. Problem Definition

Fig. 4 shows our two-layer BGA package model. Die-side ordered pin sequence (DOPS) and package-side ordered pin sequence (POPS) are the orders of I/O pins on both sides. In order to simplify the representation, we assume that each case has this pair of sequences. In reality, we can either treat it as a large pair of sequences by bending all corners to be a continuous pair of sequences, or treat four pairs of sequences for the four corners of a package. DOPS serves as input of RDL bump assignment. The corresponding RDL routing can be optimized by applying the network-flow-based [2] or the integer-linear-programming-based algorithms [3]. POPS is regarded as input of the PCB bus planner. The corresponding PCB area routing with planar bus can be readily solved by the method given in [5].

In this two-layer package model, each net (denoted as n_i), starting from DOPS, is connected to a via (denoted as v_i) on layer-1. Each via connects to exactly one ball (denoted as b_i) on layer-2. Each ball then connects to POPS using PCB escape routing. In our initial assignment, v_i and b_i are tied up as one pin (denoted with p_i)[1] and will be loosened in the post-optimization step.

[1] In this paper, we consider one signal net n_i through package and board (with assignment v_i and b_i) as pin (p_i). Therefore we use pin and net interchangeably.

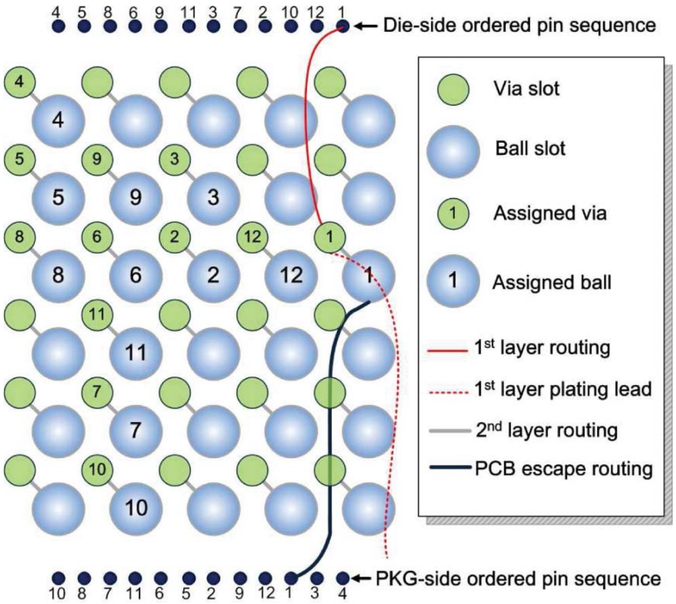


Fig. 5. Problem illustration. Each net is designated to a via and a ball; the corresponding wire planning of n_1 is plotted. v_i and b_i are tied up to find the initial solution. The numbers inside via and ball slots are the initial solution for these two ordered pin sequences.

The ballout/pin-out designation problem is to assign n_i to v_i and b_i and to generate the corresponding wire planning from given DOPS and POPS. In [10], the authors take wire length and wire congestion as their objectives; however, our design further considers package size minimization because it is important to know the tradeoffs between the wires and the die size. It is worth noting that [11] has some similarities in problem definition when the via and ball are paired for consideration; however, our problem itself is different from the simultaneous escape problem on the board. The formal definition is as follows.

Input:

1) given two sequences:
   a) DOPS;
   b) POPS.

Output:

1) ballout/pin-out designation for two-layer BGA package;
2) the corresponding wire planning (monotonic global routing) for package design and PCB escape routing.

Objectives:

1) minimize package size due to design cost (can be seen as the total number of columns used, refer to Fig. 5);
2) minimize wire congestion;
3) minimize wirelength variation/deviation.

There are six rows and five columns in the example shown in Fig. 5. The row number is counted from top to bottom, and the column number is counted from left to right. For example, the locations of p_1 and p_4 are (row 3, col 5) and (row 1, col 1), respectively. Note that each route on package layer-1 is composed of two segments: first layer routing and first layer plating lead. A plating lead is redundant for operation and is usually used to reduce fabrication cost [8]. Because of cost concerns, it is still widely used in chip packaging. According to our understanding, omitting the plating lead (especially in high-performance graphics chips) will probably double the cost. In order to solve the general case and to apply to most of the cases, we have included it in our problem definition. Plating leads are not shown in the following figures, but they are evaluated as normal wires in cost evaluation (shown in Section IV-C).

Algorithm 1 Interval-Scan

```
$i \leftarrow 1,\ j \leftarrow 2$
**Repeat:**
select net $n_i$ from DOPS and scan $I_i$
  **Repeat:**
  select net $n_j$ from DOPS and scan $I_j$
  **IF** ($I_i$ and $I_j$ go same direction and $I_i$ partially overlaps $I_j$)
  **Then** build an edge between $vt_i$ and $vt_j$
  **IF** ($I_i$ and $I_j$ go opposite direction and $I_i$ fully overlaps $I_j$)
  **Then** build an edge between $vt_i$ and $vt_j$
  increment $j$
  **Until** $I_i$ has scanned every $I_j$
increment $i$
**Until** all nets are selected
```

III. Package Pin-Out and Corresponding Wire Planning

A. Monotonic Global Routing in Wire Planning

A route from the die-side pin to via and from ball to package-side pin is called monotonic if it only intersects any straight line running parallel to the ordered pin sequence at most once. This definition is similar to the condition for monotonicity introduced in [7]. Fig. 6 shows eight routing scenarios of two-layer package routing and PCB escape routing. Only (a) and (e) are not monotonically assigned. In [8]–[10], the authors have shown that the routings on the package layer-1 are monotonic when vias are monotonicly assigned. Following the same idea, when balls are designated to be monotonic, the PCB escape routing is monotonic. In Fig. 6, three nets (n_1, n_2, and n_3) are assigned in eight different patterns. DOPS connects vias on the first layer of the package and POPS connects balls on the PCB. DOPS is given as 1, 2, 3, and POPS is given as 2, 1, 3, in which the order of n_1 and n_2 are reversed. They are called intersected nets since their flylines intersect.

Based on these scenarios, we can define the general rule of thumb for designating package pin-out and completing the monotonic global routing. Take Fig. 6(a) as an example. p_1 (pin/net 1; with via v_1 and ball b_1) is on the left of p_2, and this order is consistent with DOPS but inconsistent with POPS. When the designated column number of n_1 and n_2 is not in the same order as in DOPS or POPS, that will cause routing intersection during package layer-1 routing or PCB escape routing. To route without intersection, as shown in Fig. 6, different pin-out assignments will produce different routing results. These results are summarized as follows (row$_i$ means the row number of p_i).

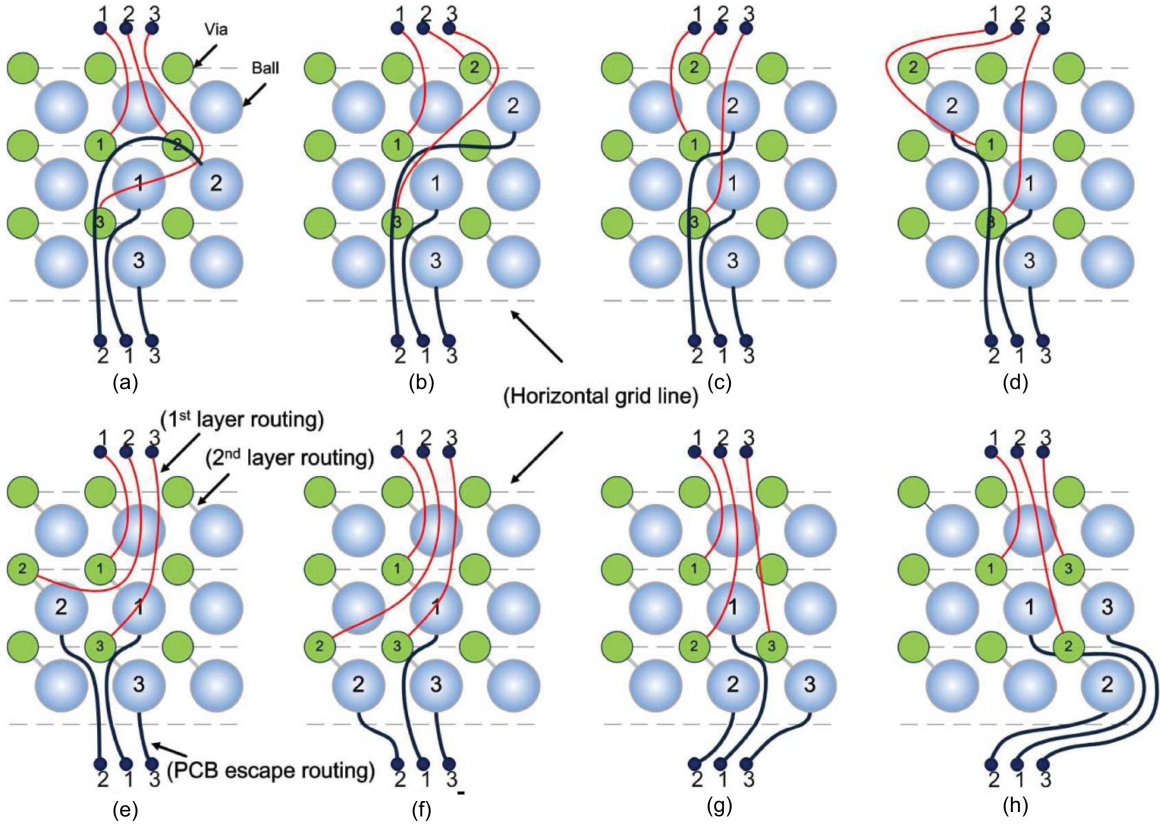


Fig. 6. Routing scenarios produced by the pins corresponding to intersected nets designated in different ways. (a) and (e) Pins are in the same row. (c) and (g) Pins are in the same column. (b), (d), (f), and (h) Pins are not in the same row or column.

Case 1 (row$_1$ = row$_2$, column$_1 \neq$ column$_2$):
In this case, p_1 and p_2 are located at the same row. To solve the routing intersection, the package layer-1 routing or PCB escape routing must be non-monotonic [see Fig. 6(a) and (e)].

Case 2 (row$_1 \neq$ row$_2$, column$_1 \neq$ column$_2$):
In this case, the assignment of p_1 and p_2 can produce monotonic routing. However, these pins will possibly be routed through more than one routing track.[2] [see Fig. 6(b), (d), (f), and (h)].

Case 3 (row$_1 \neq$ row$_2$, column$_1$ = column$_2$):
In this case, p_1 and p_2 are located at the same column. For both the package layer-1 routing and PCB escape routing, the routing results not only are monotonic but also use only one routing track. [see Fig. 6(c) and (g)].

According to these scenarios, the pin-out designation rules are defined below. We let all pairs of nets to follow.

Rule 1: To achieve monotonic routing:

a) the pins corresponding to intersected nets must not be assigned at the same row;
b) the designated column number of pins corresponding to non-interesected nets must be in the same order as in both DOPS and POPS.

Rule 2: To minimize the routing space:

a) the pins corresponding to intersected nets must be assigned at the same column;
b) the designated row number of these pins corresponding to intersected nets must be adjacent.

In order to designate package pin-out efficiently and to achieve the monotonic global routing for package design and PCB escape routing, the proposed methodology is to find the intersected relationship between nets by using an intersection graph. The pin-out assignment based on the aforementioned rules of thumb can be satisfied by applying the proposed intersecting relationship analysis.

B. Pin-Out Designation Methods for Wire Planning

In [9], a method is proposed using an inversion table to analyze the orderings of two sequences. Among all pins of each side, the intersecting relationship must be figured out to know the topology and to get the intersection graph. We use an interval diagram, which analyzes the intersection relationship of nets, to generate intersection graph. The interval diagram shown in Fig. 7 shows the intervals of the nets. For each net n_i, its corresponding interval I_i is composed of a start point s_i and a destination point d_i. s_i and d_i are represented by a small solid circle and an arrow, respectively, and they are determined by the index of n_i in DOPS and POPS, respectively.

Algorithm 1, which transforms the interval diagram into the intersection graph, is described below. Intersection graph is defined as $G_I = (V, E_I)$ and plotted in Fig. 8. $V = \{vt_i | vt_i$

[2] For the package layer-1 routing, routing track is the routing space between two column of vias. For PCB escape routing, that is the space between two columns of balls. Our objectives include wire congestion minimization, instead of limiting routing capacity of tracks.

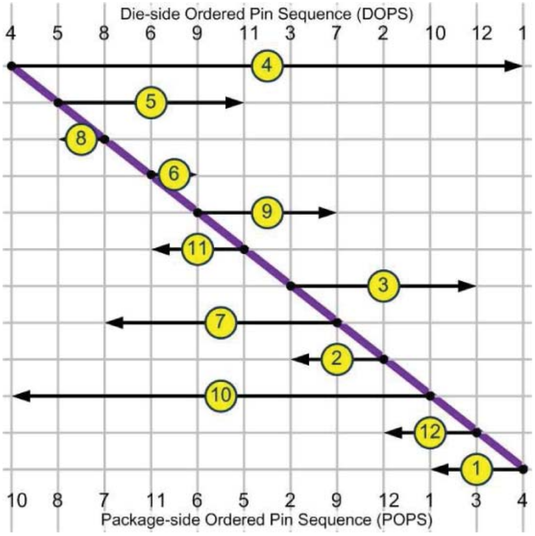

Die-side Ordered Pin Sequence (DOPS)
4 5 8 6 9 11 3 7 2 10 12 1
4
5
8
6
9
11
3
7
2
10
12
1
10 8 7 11 6 5 2 9 12 1 3 4
Package-side Ordered Pin Sequence (POPS)

Fig. 7. Interval diagram showing intervals of nets. The start and end points of arrows represent pin locations in die and package sides.

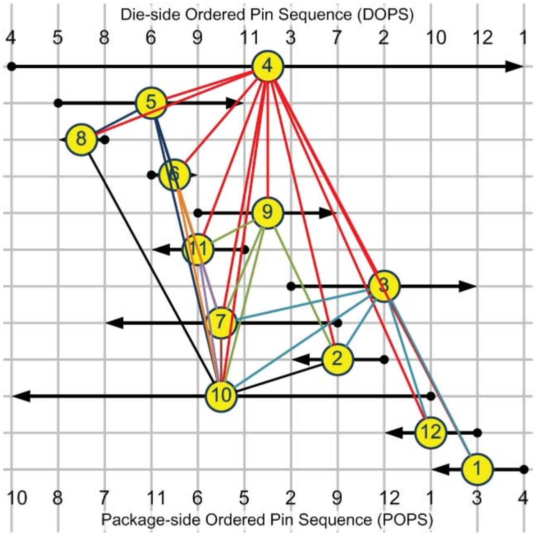

Die-side Ordered Pin Sequence (DOPS)
4 5 8 6 9 11 3 7 2 10 12 1
4
5
8
6
9
11
3
7
2
10
12
1
10 8 7 11 6 5 2 9 12 1 3 4
Package-side Ordered Pin Sequence (POPS)

Fig. 8. Intersection graph showing intersection relationship among nets. It is obtained by applying the interval-scan algorithm on the interval diagram. In intersection graph, if two intervals (an interval in interval diagram is a node in the intersection graph) intersect, an edge exists.

represents the interval I_i}. Two vertices are connected by an edge if and only if their corresponding nets intersect. To be more specific, for two nets going in opposite directions, if they are partially overlapped, which also means two flylines intersect, an edge is built. For two nets going in the same direction, an edge is built if they are fully overlapped.

We have the following lemma.

Lemma 1: If there exists an edge between two vertices and one of them is placed in some row, then the other one should be placed at the vertice row +1.

Once we have the intersection graph, the initial pin-out designation can be produced by using a simple algorithm based on the pin-out designation rules. The detailed processes are shown in Algorithm 2.

Fig. 5 shows an example of initial assignment. By using this designation algorithm, we can obtain the monotonic global routing for package design and PCB escape routing. However, in the most extreme case, where DOPS is totally reverse of POPS, we have to assign all pins to the same column. If this is not acceptable in terms of package size, we believe that the case should be unroutable and advise both sides to negotiate the pin orders. In the next section, we propose the pin-out optimization methods considering the ways to minimize the package size, routing congestion, and wirelength difference on each routing layer, which are critical concerns in chip-package–board codesign.

Algorithm 2 Initial Pin-Out Designation

```
$i \leftarrow 1,\ j \leftarrow 2$
select net $n_i$ in DOPS, assign $row_i = 1$, $column_i = 1$
**Repeat:**
select net $n_j$ in DOPS
**IF** an edge exists between $vt_j$ and $vt_i$, **Then**
  assign $row_j$ based on ***Rule 1***
  assign $column_j$ based on ***Rule 2***
**ELSE**
  assign $row_j = row_i$
  assign $column_j$ based on ***Rule 1***
$i \leftarrow j$
increment $j$
**Until** all pins in DOPS have been assigned.
```

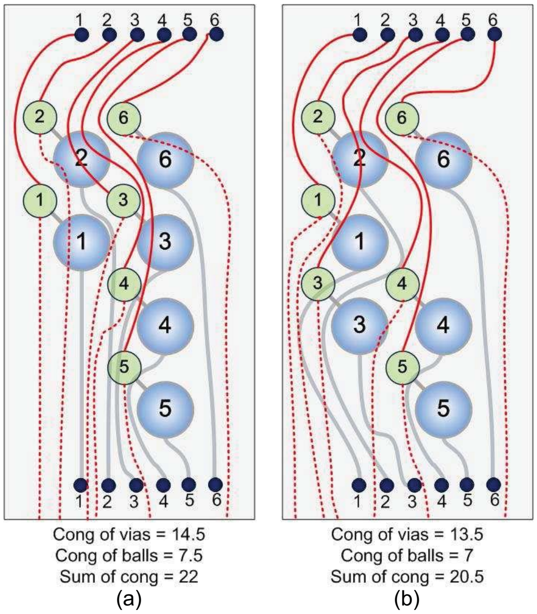


Fig. 9. Optimization for wire congestion. (a) Originally, the Cong cost is 22 for the assignment. (b) Moving via 3 and ball 3, like the assignment, can reduce Cong cost to 20.5. The formal definition of this evaluation is presented in Section IV-C.

IV. Pin-Out Optimization

A. Optimization for Individual Objectives

The optimization scheme targeting at three individual objectives is discussed in this section: package size (PS), wire congestion (Cong), and sum of length difference on each routing layer (Diff). Since our objective is to trade off the performance and cost for package design, those objectives should be justified. In high-speed digital system, length matching leads to

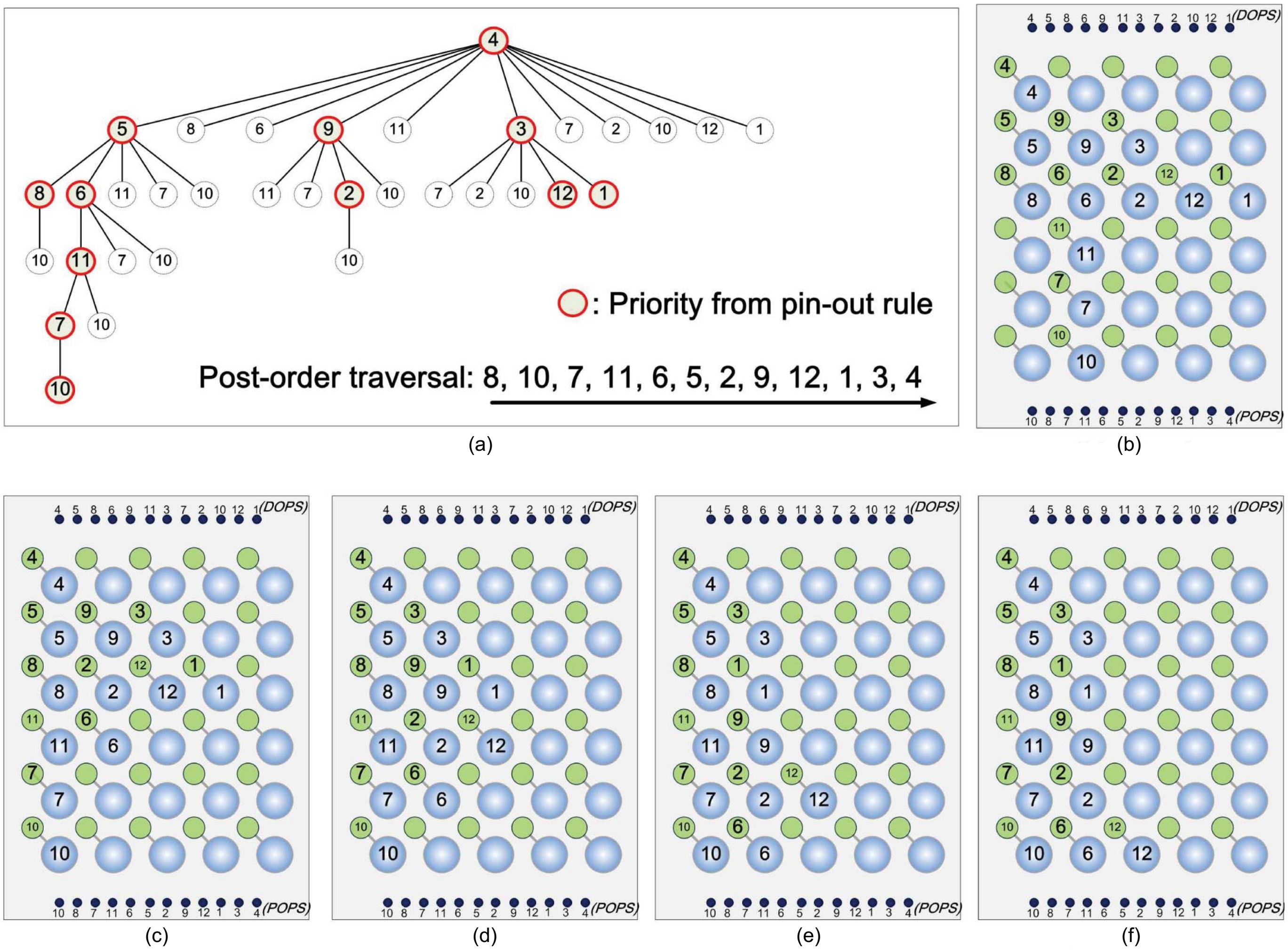

○: Priority from pin-out rule
Post-order traversal: 8, 10, 7, 11, 6, 5, 2, 9, 12, 1, 3, 4
4 5 8 6 9 11 3 7 2 10 12 1 (DOPS)
10 8 7 11 6 5 2 9 12 1 3 4 (POPS)
(a)
(b)
(c)
(d)
(e)
(f)

Fig. 10. Optimization for package size. (a) Priority tree generated from Fig. 5. (b)–(f) Status of movement. The number of columns used is decreased from 5 to 3.

the minimum signal skew and noise, which are critical factors for differential signaling. Besides, the maximum wirelength will be reduced while we minimize the package size. This is the reason why we use length difference and package size as our cost metrics. Considering the systematic design of chip–package–board, we can adjust the weight of factors in our cost metric to optimize the system.

Cong minimization is achieved by two intuitive methods. One is to equally distribute routing channels: when pins are constrained to be at the same row, intuitively, averaged routing channel distribution minimizes wire congestion. For example, when there are four columns and two pins, it is best to assign p_1 and p_2 to column$_2$ and column$_4$, respectively. The other method is to change the column number of pins. Looking at Fig. 9, moving p_3 out of the original row relieves Cong a lot. It is obvious that focusing on Cong minimization possibly enlarges pin-block size, but the proposed general optimization scheme can still find the assignments that decrease Cong while preserving PS.

Diff minimization is to minimize the variation in wirelength for each layer. Length differences for each layer are considered separately if they are not uniform interconnects. Note that the sum of routes is longer on the package layer-1 because plating leads are required in low- or medium-cost packaging. Rather than minimizing all wires altogether, minimizing the longest wires is sufficient because they often dominate the Diff term.

PS minimization[3] can be obtained if the pins are moved in a certain order iteratively. We can obtain the priority tree from the intersection graph in the following example. Since intersection graph is the solution space of all legal (not violating rules 1 and 2) solutions, the priority tree is a subset of the intersection graph; the priority tree in Fig. 10(a) represents the example initial solution in Fig. 5. We can spread out the intersection graph, in Fig. 8, there are edges for nodes 4 and 8, and for nodes 5 and 8; therefore in Fig. 10(a), node 8 is the child of nodes 4 and 5. We choose the node placed in the first row (node 4 in this example), we can pick any one if there are many. The order is then generated by post-order traversal of this priority tree. For instance, it is 8, 10, 7, 11, 6, 5, 2, 9, 12, 1, 3, 4 in Fig. 8 and the tree in Fig. 10(a).

[3] PS minimization here does not include P/G pins. For signal-limit package, less P/G can be acceptable. In this paper, we focus on package routing planning for cost-effective package design. Signal and P/G pin coplanning would be one of our future works.

The pins are moved sequentially in this order and obey the direction priority (go left > go bottom-left > go down). The direction priority is that we prefer decreasing column count (moving left) than row count (moving bottom). The reason of obeying this order is that the children should be moved before the parents. If the priority of moving children is higher than that of moving parents, children can always stay left and/or bottom corners than the parents. Fig. 10(b)–(f) illustrates each step of the procedure in minimizing PS. For each step, only one pin is moved at a time as long as there exists an empty via/ball slot, and all pins are moved in order. Each pin stops moving if it touches the boundary, thus the number of rows cannot be increased. In the initial solution, the number of rows is minimum, which is determined by the depth of intersection tree shown in Fig. 10(a). In order to preserve monotonic assignment, pins that cross each other cannot be placed in the same row. Thus, to minimize package size is to try to minimize the number of columns.

B. Unified Cost Optimization

The proposed optimization scheme is to: 1) select one pin/via/ball which costs most; 2) search for its legal neighbors; and 3) perform operations between the pin/via/ball and its legal neighbors. It is important to honor the legality in order to keep the assignment monotonic. The costs of the via grid array (cVGA) and the cBGA are summed up. So an optimization step that merits cVGA may demerit cBGA.

We use the following heuristics to find better solutions in moving pin/via/ball. Note that Cong, Diff, and PS are normalized for a fair comparison with each other.

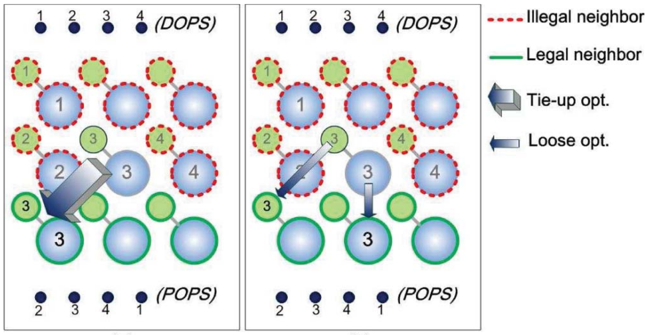
1) Greedy method: Starting from initial solution, only downhill searches are accepted. The method keeps moving the most expensive pin/via/ball to its less expensive neighbors. It is useful when there is one pin/via/ball which contributes a lot in cost. However, it is not suitable when there is a group of pins/vias/balls which should be optimized simultaneously.
2) Lowest partial cost (LPC) method: Serial of moves are first accepted to escape local optima. Moves are performed whose accumulated sum of costs is a minimum. The first move is relatively important because the quality of all the following moves depends on it.

The optimization scheme is conducted in two stages: tie-up optimization, followed by loose optimization. Each pair v_i and b_i are tied up as p_i to search for a global optimum in the first stage, and then they are loosened to search for local optima in the second stage. Fig. 11 shows an optimization step for n_3. p_3 attempts to decrease the cost by exploring its neighbors in Fig. 11(a), while v_3 and b_3 search to decrease their own cost separately.

C. Cost Evaluation

In the unified cost optimization, the cost function is defined as follows:

$$\text{Cost}_{vi/bi} = \alpha \times \text{Cong}_{vi/bi} + \beta \times \text{Diff}_{vi/bi} + \gamma \times \text{PS}_{vi/bi} \quad (1)$$



(a) (b)

Fig. 11. Post optimization. We first tie up v_3 and b_3 for global search, and then loosen this constraint so that they can separately find other local solutions to reduce the cost. (a) Tie-up optimization. (b) Loose optimization.

where $\text{Cost}_{vi/bi}$ indicates the cost of a via v_i or a ball b_i, and α, β, and γ are user-defined parameters. Each via/ball has a cost composed of Cong, Diff, and PS. And the total cost is the sum of the cost of all vias/balls. Here we define the cost of three objectives separately.

The cost of wire congestion of two via/balls is defined as

$$\text{Cong}_{vi/bi} = \frac{\text{no. of wire}_\text{l}}{\text{no. of channel}_\text{l}} + \frac{\text{no. of wire}_\text{r}}{\text{no. of channel}_\text{r}} \quad (2)$$

where no of wire$_{l/r}$ denotes the number of wires that lie on the left/right, and no of channel$_{l/r}$ denotes the number of routing channels on the left/right, shown in Fig. 12. Note that the plating leads are metal wires, so they also contribute to Cong.

The cost of length difference is defined as

$$\text{Diff}_{vi/bi} = |\text{dist}(v_i/b_i) - \text{dist}(\text{avg})| \quad (3)$$

where $\text{dist}(v_i/b_i)$ denotes the Manhattan distance of the via/ball and dist(avg) denotes the averaged Manhattan distance of all vias/balls. Since the monotonic assignment can guarantee monotonic routing, using Manhattan distance to estimate wirelength is sufficient.

The cost of PS is defined as

$$\text{PS}_{\frac{vi}{bi}} = \begin{cases} \frac{1}{\left[\frac{\text{no. of v}}{b}\right]_V} + \frac{1}{\left[\frac{\text{no. of v}}{b}\right]_H} \times W \times L, & \text{if } \left[\frac{\text{no. of v}}{b}\right]_{\frac{V}{H}} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $[\text{no. of v/b}]_\text{V}$ and $[\text{no. of v/b}]_\text{H}$ denote the number of vias/balls that lie on the vertical and horizontal boundary, respectively, W and L denote number of columns and rows of package size, respectively, shown in Fig. 12.

V. Experimental Results

Our algorithm is implemented using C++ on a 3.0-GHz Intel Xeon Quad Core Processor 5160 PC under the Linux operating system. In the following tables, cVGA denotes the total cost of via grid array, cBGA denotes the total cost of ball grid array, and Sum is the sum of cVGA and cBGA

$$\text{cVGA} = \sum_{i=0}^{n} \text{Cost}_{vi} \quad (5)$$

TABLE I

COMPARISON OF THE RESULTS OF [10] AND OUR POST-OPTIMIZATION SCHEME WITH OUR PROPOSED INITIAL SOLUTION AND OUR COST METRICS. IN OUR INITIAL SOLUTIONS, CONG, DIFF, AND PS ARE ALL EVALUATED AS 1.00. THE INITIAL SOLUTIONS OUTPERFORM [10] BECAUSE MOST COSTS OF [10] ARE GREATER THAN 1.00. GREEDY METHOD CAN FURTHER IMPROVE THE INITIAL SOLUTIONS BY 22% ON AVERAGE. THE RESULTS OF BENCH-1 FOR [10] ARE NOT AVAILABLE

	[10] (opt. for cVGA only)									
	VGA			BGA			Imp.%			
	Cong	Diff	PS	Cong	Diff	PS	Cong	Diff	PS	Sum
Bench-2	2.09	1.41	0.93	1.61	1.34	0.47	–85%	–38%	30%	–31%
Bench-3	1.70	4.21	1.30	1.50	1.12	1.30	–60%	–166%	–30%	–85%
Bench-4	1.59	4.03	0.47	1.21	1.60	0.47	–40%	–182%	–53%	–56%
Bench-5	1.09	3.63	1.08	0.98	1.21	1.62	–4%	–142%	–35%	–42%
Avg.	1.62	3.32	0.94	1.33	1.32	0.96	–47%	–132%	5%	–54%
	Greedy method full mode									
	VGA			BGA			Imp.%			
	Cong	Diff	PS	Cong	Diff	PS	Cong	Diff	PS	Sum
Bench-2	0.85	0.14	0.95	0.77	0.23	0.95	19%	81%	5%	35%
Bench-3	0.75	0.30	1.06	0.80	0.31	1.00	22%	69%	–3%	30%
Bench-4	0.95	0.89	1.00	1.05	0.55	1.00	0%	28%	0%	9%
Bench-5	0.87	0.84	1.00	0.94	0.58	1.00	9%	29%	0%	13%
Avg.	0.86	0.54	1.00	0.89	0.42	0.99	13%	52%	1%	22%

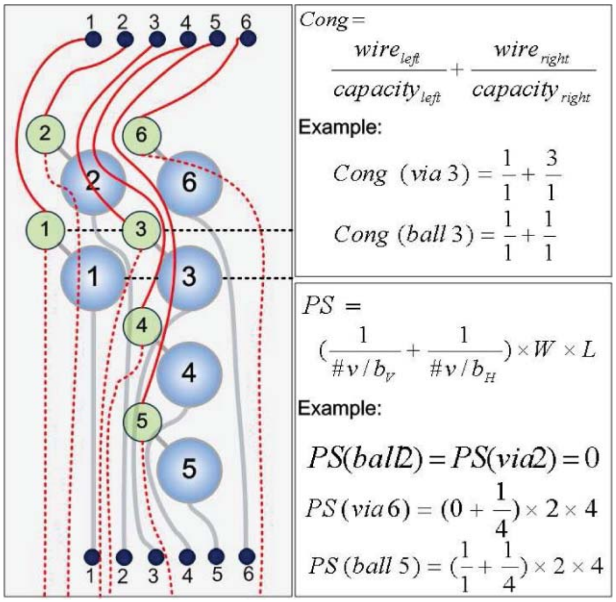

1 2 3 4 5 6
2
6
1
3
1
3
4
4
5
5
1 2 3 4 5 6
Cong=
wire_left / capacity_left + wire_right / capacity_right
Example:
Cong (via 3) = 1/1 + 3/1
Cong (ball 3) = 1/1 + 1/1
PS =
(1/#v/b_V + 1/#v/b_H) × W × L
Example:
PS(ball2) = PS(via2) = 0
PS (via 6) = (0 + 1/4) × 2 × 4
PS (ball 5) = (1/1 + 1/4) × 2 × 4

Fig. 12. Cost evaluation (Cong and PS) for via and ball. For Cong (via 3), there is one wire (dotted red) on the left of via3; 3 wires (2 solid red and 1 dotted red) on the right of via3, so Cong(via3) is calculated as $1/1 + 3/1$. For PS (via6), column 2 is the right-most column that contributes to package size; there are four vias on column 2, so PS (via6) is $(0 + 1/4) \times 2 \times 4$.

$$\text{cBGA} = \sum_{i=0}^{n} \text{Cost}_{bi} \tag{6}$$

$$\text{Sum} = \text{cVGA} + \text{cBGA}. \tag{7}$$

Two optimization schemes, Greedy and LPC, are implemented and tested in two modes: tie-up and full. tie-up indicates that, for n_i, v_i, and b_i are tied up as p_i to optimize simultaneously. full indicates that, after having conducted tie-up, p_i is loosened to perform optimization for vias and balls separately. The proposed initial solution is generated by initial pin-out designation algorithm proposed in Section III. Recall (1), the cost of v_i/b_i is composed of $\text{PS}_{\text{vi/bi}}$, $\text{Diff}_{\text{vi/bi}}$, and $\text{Cong}_{\text{vi/bi}}$. In the experiments, the coefficients α, β, and γ are normalized to the initial solution and defined as

$$\alpha \times \sum_{i=0}^{n} \text{Cong}_{vi/bi} = 1 \tag{8}$$

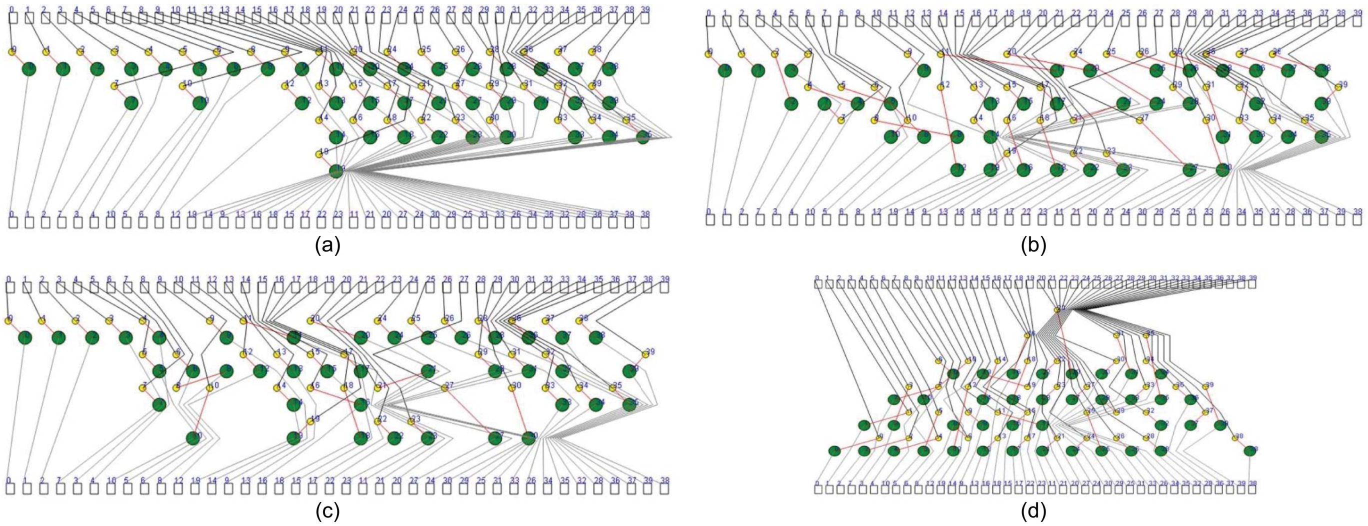
$$\beta \times \sum_{i=0}^{n} \text{Diff}_{vi/bi} = 1 \tag{9}$$

$$\gamma \times \sum_{i=0}^{n} \text{PS}_{vi/bi} = 1. \tag{10}$$

Therefore, $\text{cVGA}_{\text{initial\_sol}} = \text{cBGA}_{\text{initial\_sol}} = 3.00$ and $\text{Sum}_{\text{initial\_sol}} = 6.00$.

In order to show the effectiveness of our package wire planning, we compare with [10] in our specified congestion perspective. Because of the different perspectives in the focus, we use difference cost metrics, and here we detail the difference. Both [10] and the proposed methodology adopt the same two-layer package model. However, [10] assigns its initial solution randomly and performs optimization for cVGA only. Their cost evaluation considers total wirelengh minimization, while we focus on length-variation minimization. Besides, their package size is given as input, while ours can decide the tradeoff between package size and RA. The authors of [10] consider both vertical and horizontal congestion, while we consider horizontal congestion only. Methodologies in [10] optimize for their cost function such as congestion. Actually, the approach of [10] and our approach do not target the same problem; we take [10] as comparison only because it is the most similar, to the best of our knowledge.

In Table I, we test the proposed methodology in four industrial cases; the results of bench-1 for [10] are not available. Note that most costs of [10] are greater than 1.00, which


(a)

(b)

(c)

(d)

Fig. 13. Experimental results for bench-3. (a) Proposed initial solution. (b) Result of greedy method. (c) Result of LPC method. (d) Result of [10].

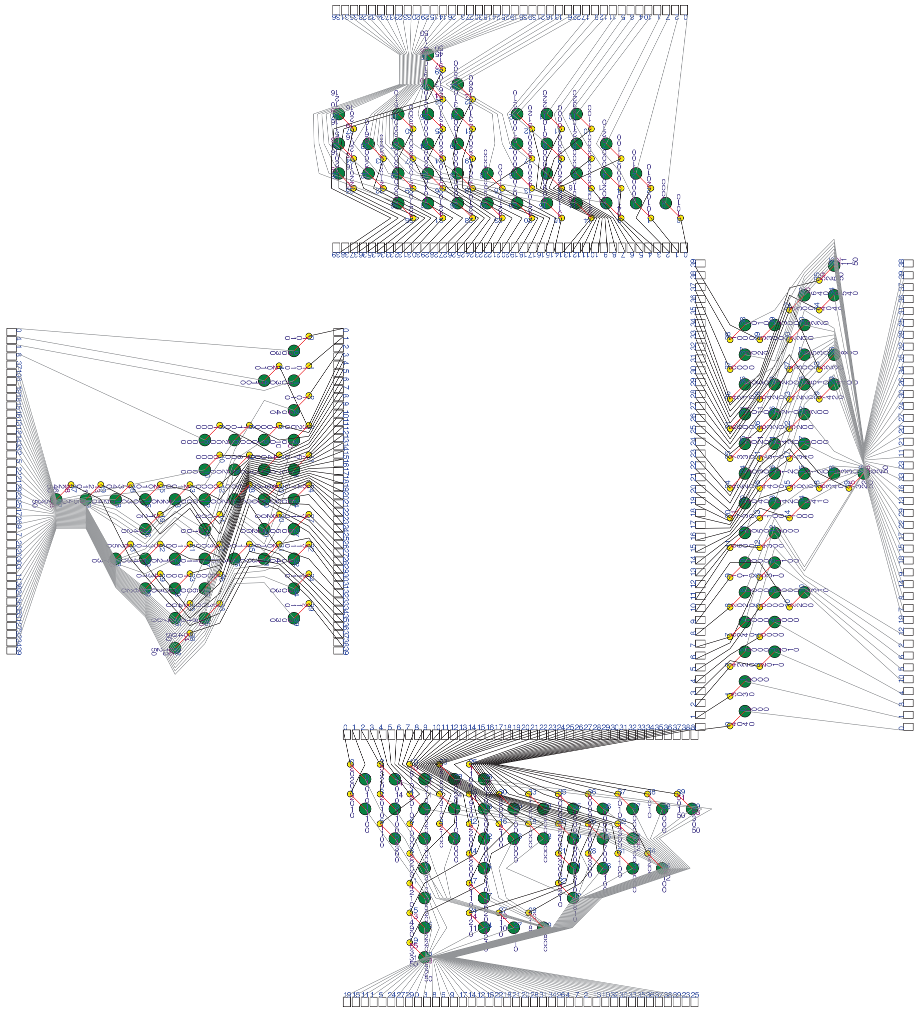


Fig. 14. Experimental results for a full case, more than 100 pins.

means that they are more expensive than our initial solution. The proposed initial solution improves 54% on average, compared to [10]. The initial solutions can be further improved by 22% on average after applying Greedy method in full mode.

Table II shows the results of two optimization schemes Greedy and LPC. Similar behaviors are observed for most of the results. They improve the initial solutions by 16% in the tie-up mode. This can be further optimized in the full

TABLE II

COMPARISON OF DIFFERENT METHODS AND MODES WITH THE INITIAL SOLUTIONS. TAKING BENCH-1 AS AN EXAMPLE, THE RESULTS OF GREEDY METHOD IN TIE-UP MODE IMPROVES DIFF AND PS BY 53% AND 25%, RESPECTIVELY, BUT IT WORSENS CONG BY 27%. OUR POST-OPTIMIZATION SCHEMES SHOW THAT THE IMPROVEMENTS COMPARED WITH THE PROPOSED INITIAL SOLUTION ARE ON AVERAGE GREATER THAN 20%

	Greedy method							
	Tie-up mode				Full mode			
	Cong	Diff	PS	Sum	Cong	Diff	PS	Sum
Bench-1	–27%	53%	25%	17%	–12%	49%	25%	21%
Bench-2	7%	73%	5%	30%	19%	81%	5%	35%
Bench-3	13%	43%	0%	19%	22%	69%	–3%	30%
Bench-4	–1%	11%	0%	3%	0%	28%	0%	9%
Bench-5	13%	14%	0%	9%	9%	29%	0%	13%
Avg.	1%	39%	6%	16%	8%	51%	5%	22%
	LPC method							
	Tie-up mode				Full mode			
	Cong	Diff	PS	Sum	Cong	Diff	PS	Sum
Bench-1	–27%	53%	25%	17%	–12%	49%	25%	21%
Bench-2	12%	73%	5%	30%	18%	78%	5%	33%
Bench-3	13%	43%	0%	19%	18%	58%	–3%	24%
Bench-4	–1%	11%	0%	3%	–1%	26%	0%	8%
Bench-5	12%	15%	0%	9%	8%	29%	0%	12%
Avg.	2%	39%	6%	16%	6%	48%	5%	20%

mode to give a final improvement of 20%–22% on average. Note that the initial assignment of bench-1 is shown previously in Fig. 5. The execution time for all experiments is less than 1 s.

The results of bench-3 are plotted in Fig. 13. Fig. 13(a) shows the proposed initial assignment in which all wires are planned monotonically; Fig. 13(b) and (c) show the results of post-optimization using Greedy and LPC methods, respectively. They have similar patterns with slightly different assignments. The cost of (b) is lower than that of (c) by 6% because the greedy method can find better solution in the loose mode. This shows that the cost of BGA benefits more than that of VGA from loose optimization. Fig. 13(d) is one of the experimental results in [10], which shows a smaller package size; however, it is more congested and has larger variation in wirelength. Fig. 14 shows the full case for bench-3. If we consider the pin-block idea in our previous work [1], we can solve several hundreds or thousands of I/O pins.

VI. CONCLUSION

In order to address the long-existing problem in slow turnaround between design, package, and system houses, we defined a new subproblem that helps the fast estimation of wire planning in chip–package–board codesign. Core designers can specify the preferred I/O pad ordering, and system designers can specify the preferred bump pin-out designation. We can efficiently analyze if the preferences of both sides accommodate each other before performing RDL routing and substrate routing. We also considered the essential concerns in package design, such as routing congestion, package size, and length deviation among all nets. The results show the method's effectiveness and efficiency. We plan to work on SI and high-speed designs in future work.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for providing valuable comments, which greatly improved this paper.

REFERENCES

[1] R.-J. Lee and H.-M. Chen, "Fast flip-chip pin-out designation respin for package-board codesign," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 8, pp. 1087–1098, Aug. 2009.

[2] J.-W. Fang, I.-J. Lin, Y.-W. Chang, and J.-H. Wang, "A network-flow-based RDL routing algorithmz for flip-chip design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 8, pp. 1417–1429, Aug. 2007.

[3] J.-W. Fang, C.-H. Hsu, and Y.-W. Chang, "An integer-linear-programming-based routing algorithm for flip-chip designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 1, pp. 98–110, Jan. 2009.

[4] M. M. Ozdal and D.-F. Wong, "Algorithms for simultaneous escape routing and layer assignment of dense PCBs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 8, pp. 1510–1522, Aug. 2006.

[5] H. Kong, T. Yan, and D.-F. Wong, "Automatic bus planner for dense PCBs," in *Proc. ACM/IEEE Design Autom. Conf.*, Jul. 2009, pp. 326–331.

[6] L. Luo and D.-F. Wong, "Ordered escape routing based on Boolean satisfiability," in *Proc. Asia South Pacific Design Autom. Conf.*, 2008, pp. 244–249.

[7] Y. Tomioka and A. Takahashi, "Monotonic parallel and orthogonal routing for single-layer ball grid array packages," in *Proc. Asia South Pacific Design Autom. Conf.*, 2006, pp. 24–27.

[8] Y. Kubo and A. Takahashi, "A global routing method for 2-layer ball grid array packages," in *Proc. Int. Symp. Phys. Design*, 2005, pp. 36–43.

[9] S.-S. Chen, J.-J. Chen, T.-Y. Lee, C.-C. Tsai, and S.-J. Chen, "A new approach to the ball grid array package routing," *Trans. IEICE*, vol. E82-A, no. 11, pp. 2599–2608, 1999.

[10] Y. Tomioka and A. Takahashi, "Routability driven modification method of monotonic via assignment for 2-layer ball grid array packages," in *Proc. Asia South Pacific Design Autom. Conf.*, 2008, pp. 238–243.

[11] L. Luo, T. Yan, Q. Ma, D. F. Wong, and T. Shibuya, "A new strategy for simultaneous escape based on boundary routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 2, pp. 205–214, Feb. 2011.

Ren-Jie Lee (S'07) received the M.S. degree in electronics engineering from Feng Chia University, Taichung, Taiwan, and the Ph.D. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2000 and 2010, respectively.

He was a Project Manager with Silicon Integrated Systems Corporation, Hsinchu, from 2000 to 2006. He is currently a Senior Engineer with the Design Technology Engineering Division, NOVATEK Microelectronics Corporation, Hsinchu. His current research interests include beyond die-integration and package/hybrid/board design automation, including chip-package-board codesign, system-in-package design, and analysis and optimization beyond the die.

Hung-Ming Chen (M'03) received the B.S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the M.S. and Ph.D. degrees in computer sciences from the University of Texas at Austin in 1998 and 2003, respectively.

He is currently an Associate Professor with the Department of Electronics Engineering, National Chiao Tung University. His current research interests include physical design automation in digital and analog circuits, beyond-die integration (off-chip EDA), and 3-D IC design methodology.

Dr. Chen has served as the Technical Program Committee Member of the ACM/IEEE ASP-DAC, IEEE SOCC, VLSI-DAT, and ACM ISPD.

Hsin-Wu Hsu received the B.S. degree in electronics engineering and the M.S. degree with the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan, in 2009 and 2011, respectively, and the M.S. degree from Universite Paris-Sud 11, Paris, France, in 2010.

He is currently an Engineer with Taiwan Semiconductor Manufacturing Company Ltd., Hsinchu. His current research interests include routing and packaging CAD.

Dr. Hsu has been awarded for his master thesis from Taiwan IC Design Society.