

# An $O(N \log N)$ Algorithm for Region Definition Using Channels/Switchboxes and Ordering Assignment

JIN-TAI YAN and PEI-YUNG HSIAO

Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

(Received September 25, 1993, Revised August 20, 1994)

For a building block placement, the routing space can be further partitioned into channels and switchboxes. In general, the definition of switchboxes releases the cyclic channel precedence constraints and further yields a safe routing ordering process. However, switchbox routing is more difficult than channel routing. In this paper, an  $O(N \log N)$  region definition and ordering assignment (RDAOA) algorithm is proposed to minimize the number of switchboxes for the routing phase, where  $N$  is the number of vertices in a channel precedence graph. Several examples have been tested on the proposed algorithm, and the experimental results are listed and compared.

**Key Words:** *Building Block Placement, Region Definition, Ordering Assignment, Switchbox*

## 1 INTRODUCTION

In VLSI layout design, most of the layout problems[1] have been proven to be NP-complete, and heuristic approaches have been applied to reduce the complexity of those problems. Thus, many efficient algorithms have been proposed to obtain near-optimal solutions in reasonable time complexity. Basically, these layout problems are classified into the following main subproblems: placement, global routing, region definition and ordering assignment (RDAOA), detailed routing, and compaction. In this paper, we consider the RDAOA problem and propose an efficient algorithm for the problem.

Consider the building block placement shown in Figure 1 and suppose that regions A, B, C, and D are all to be routed as channels. From the viewpoint of channel definition, the terminals in region B are not fixed, so region A must be routed before region B. For the same reason, region B must be routed before region C, region C must be routed before region D, and region D must be routed before region A. As a result, the iterative phenomenon of the precedence relation will construct a precedence cycle, and the cyclic constraint is named a cyclic channel precedence constraint[2]. Clearly, as a building block placement is with any cyclic channel precedence constraint, the placement will not be a slicing placement and the routing space will not be only decomposed into linear channels. In general, the cyclic channel prece-

dence constraints are released by the definition of switchboxes or L-shaped channels.

The solutions for breaking cyclic channel precedence constraints in the RDAOA problem have been studied for many years. Many various approaches have been proposed and discussed extensively in previous papers. For example, Otten[3] restricts the acceptable placements of slicing structures in order to avoid cyclic channel precedence constraints. Hence, it is clear that there is no cyclic channel precedence constraint in the given placement. On the other hand, a given placement may be modified to avoid cyclic channel precedence constraints by converting a nonslicing structure into a slicing structure. For example, Chiba[4] perturbs the placement to convert a non-slicing structure into a slicing structure, and Kimura[5] shrinks the placed modules to perform the same conversion. However, these techniques are to avoid the appearance of cyclic channel precedence constraints and not to break cyclic channel precedence constraints in a building block placement.

In recent years, the cyclic channel precedence constraints in a placement have been broken by introducing switchboxes[6-8] or by combining linear channels into L-shaped channels[2][9-10]. From the viewpoint of routability, the width of a switchbox or an L-shaped channel is fixed and the routing completion is not guaranteed. In addition, switchbox routing or L-shaped channel routing is difficult than linear channel routing.

Hence, it is important for the completion of the routing phase to reduce the number of switchboxes or L-shaped channels. As a result, the purpose of the RDAOA problem will be to study how to introduce and minimize the number of switchboxes or L-shaped channels to break all of the cyclic channel precedence constraints in the building block placement. Recently, For the reduction of the number of switchboxes, Cai and Wong[11] propose a graph-theory approach that makes use of an  $O(N^3)$  algorithm for computing minimum clique covers of triangulated graphs to minimize the number of switchboxes in 1991. However, the algorithm is too complicated and its time complexity is too inefficient. According to the properties of cyclic channel precedence constraints, we will propose an efficient algorithm for the RDAOA problem.

For a cyclic channel precedence constraint, we can refer again to the building block placement shown in Figure 1. First, the height of region A can be estimated and fixed according to the routability consideration. The topological wiring paths on the junctions terminals between region A and region B will be generated and fixed in the global routing phase[12]. Then region B, region C and region D can now be routed sequentially as linear channels in that order. Subsequently, region A becomes a switchbox with fixed terminals on three sides and is routed by a switchbox router. As mentioned above, the cyclic channel precedence constraint will be broken by introducing a switchbox. By the same process, all the cyclic channel precedence constraints in the given placement will be broken by the definition of several switchboxes.

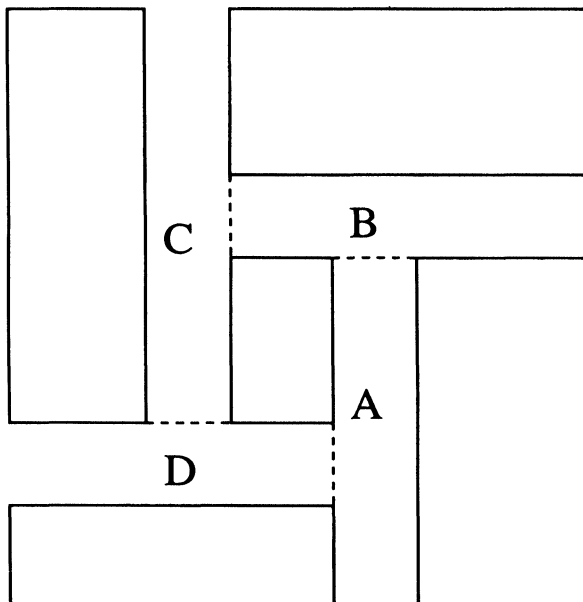


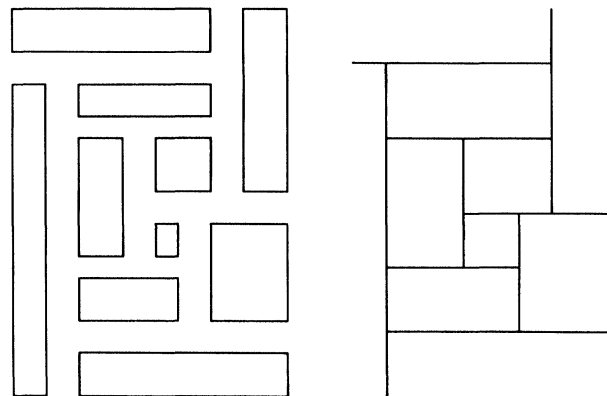
FIGURE 1 A cyclic channel precedence constraint.

According to the previous description in the RDAOA problem, all the building blocks can be located on fixed positions by a macro-cell placement algorithm. In general, the routing space between any pair of adjacent blocks can be applied to route all of the nets in the routing phase. In order to guarantee the completion of the routing phase, the routing space must be partitioned into channels and minimum switchboxes, and the routing channels and switchboxes must be further assigned in a safe order. In this paper, we propose an  $O(N \log N)$  algorithm to partition the routing space into channels and minimum switchboxes and assign their routing ordering of these channels and switchboxes.

## 2 PRELIMINARIES AND DEFINITIONS

For the RDAOA problem, the following assumptions must be assigned to the proposed algorithm. Firstly, assume that all of the building blocks belong to rectangular modules. The routing space between any pair of adjacent blocks is represented by one horizontal or vertical line segment. Thus, the intersection points of the line segments represent the junction routing areas. From the geometrical relation of line segments and intersection points, all of the line segments and intersection points will construct a floorplan graph in the building block placement. Such a floorplan graph can be generated from a building block placement by a transformation algorithm[3]. The floorplan graph can be further applied to define the channels and switchboxes and assign their routing order. In Figure 2 (a)(b), a building block placement and its related floorplan graph are shown.

Secondly, as a floorplan graph has been generated from a building block placement, it is necessary to focus on two conditions of empty rooms and “+” type junctions[2]. Fortunately, the conditions of empty rooms and



(a) A given placement

(b) The floorplan graph

FIGURE 2 A building block placement and its floorplan graph.

“+” type junctions have been studied and solved in the previous related papers. Based on these solutions, whenever there exists any empty room in the floorplan graph, the room will be removed by recursively removing one wall segment from each empty room until none are present in the floorplan graph[13]. Thus, it may be assumed that all of the empty rooms in the floorplan graph have been removed for the RDAOA problem. On the other hand, for the “+” type junctions, some algorithms were proposed to split one “+” type junction into two “T” type junctions[14]. In general, those splitting algorithms always remove all the “+” type junctions successfully in the floorplan graph by creating two “T” type junctions. Therefore, it may be assumed that the “+” type junctions in the floorplan graph have been separated into two “T” type junctions by a splitting algorithm for the RDAOA problem.

Based on the assumptions as mentioned above, a floorplan graph for the RDAOA problem only consists of horizontal and vertical line segments. Basically, each horizontal or vertical line segment in the floorplan graph can be represented as a linear channel. As two rows of terminals on one channel are fixed, the channel will be successfully routed by a channel router. In general, each “T” type junction in the floorplan graph consists of one horizontal channel and one vertical channel, and the base channel of the “T” type junction must be routed before the top channel of that. Therefore, the “T” type junction can be solved by a precedence routing relation. Based on the routing precedence relation, we can formally define a channel precedence graph from a floorplan graph and the related definitions can be discussed as follows:

**Definition 1:** For a floorplan graph,  $F$ , a channel precedence graph,  $G(F) = (V, A)$ , is a directed graph defined as follows: Each line segment in the floorplan graph is represented as a vertex of  $V$ . Each directed edge  $(u, v)$  is in  $A$  if and only if the line segments corresponding to  $u$  and  $v$  form a “T” type junction in the floorplan graph, and the one corresponding to  $u$  is the base of the “T” type junction.

In Figure 3(a), the floorplan graph of a building block placement is shown, where  $v_i$  denotes the  $i$ -th vertical line segment from the left, and the  $h_j$  denotes the  $j$ -th horizontal line segment from the top. The channel precedence graph  $G$  of Figure 3(a) is illustrated in Figure 3(b), where  $V$  is  $\{v_1, v_2, v_3, v_4, v_5, v_6, h_1, h_2, h_3, h_4, h_5\}$  and  $A$  is  $\{(v_1, h_1), (v_2, h_2), (v_3, h_2), (v_3, h_4), (v_4, h_3), (v_5, h_3), (v_6, h_5), (h_1, v_2), (h_2, v_1), (h_2, v_5), (h_3, v_3), (h_3, v_6), (h_4, v_1), (h_4, v_4), (h_5, v_4)\}$ .

From the construction of a channel precedence graph, if the graph is acyclic, the routing space will be decomposed into horizontal and vertical linear channels, and the order of these channels will be decided by the

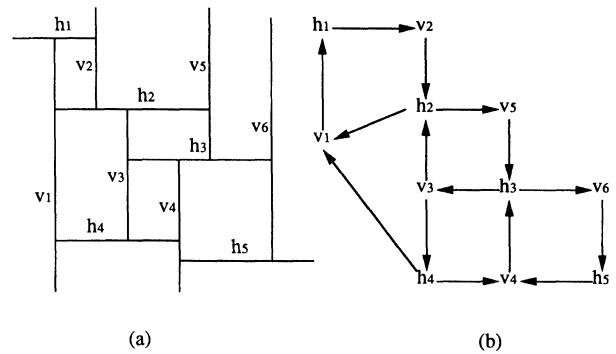


FIGURE 3 (a) A given floorplan graph. (b) The channel precedence graph.

topological sorting in the graph. It is clear that the RDAOA problem can be solved by a topological sorting algorithm and no switchbox is assigned in the building block placement. Hence, it is unnecessary to minimize the number of switchboxes in the RDAOA problem. On the other hand, if the graph is cyclic, the RDAOA problem will not be solved by a topological sorting algorithm, and some vertices in the channel precedence graph must be defined as switchboxes to break all the cyclic channel routing constraints. Therefore, it is necessary for the RDAOA problem to remove a feedback vertex set from the channel precedence graph. The feedback vertex set in a directed graph will be defined as following:

**Definition 2:** A feedback vertex set  $S$  in a directed graph  $G(V, A)$  is a vertex subset of  $V$ . The vertices in  $S$  removed from  $G$  result in an acyclic directed graph. A minimum feedback vertex set  $S_{\min}$  is a feedback vertex set with minimum cardinality.

From the viewpoint of the RDAOA problem, if a channel precedence graph is cyclic, some vertices will be removed from the graph to generate a new acyclic channel precedence graph. As the removed vertices are defined as switchboxes, all the cyclic channel precedence constraints in the placement will be successfully broken and the vertices in the remaining acyclic channel precedence graph will be further defined as channels and ordered by a topological sorting. Therefore, the RDAOA problem in a building block placement will correspond to the minimum feedback vertex set (MFVS) problem in a channel precedence graph.

In general, the MFVS problem remains NP-hard[15] for a general directed graph. However, a channel precedence graph contains available graphical properties, and these properties can be applied to solve the MFVS problem. These graphical properties will be described in the following:

**Lemma 1:** Let  $G$  be a channel precedence graph for a building block placement,  $G$  has the following graphical properties:

- (1) Planar.
- (2) Bipartite.
- (3) Maximum out-degree is equal to 2.
- (4) If  $|V| \geq 4$ , then there are at least four vertices whose out-degree is not more than 1.
- (5) The length of the smaller cycle, the minimal cycle, is 4.

**Proof:** The statement (1)–(4) have been proven in Cai and Wong’s paper[11]. Furthermore, the proof of statement (5) can be explained as follows: one edge in any cycle is constructed by the “T” relation between one vertical segment and one horizontal segment. Hence, the number of vertical segments is equal to the number of horizontal segments in a cycle. Since the length of no cycle is 2, the length of the smaller cycle is 4. **Q.E.D.**

According to the topological position of building blocks, there will exist some minimal cycles in a channel precedence graph. From another viewpoint, each vertex in the graph may be located on at most four minimal cycles. For any vertex in the graph, the number of minimal cycles containing the vertex will be used as heuristic to minimize the number of switchboxes in the RDAOA problem.

For assigning the number of minimal cycles  $N_{\text{cycle}}$  for each vertex, it is necessary to detect all the minimal cycles in a channel precedence graph. Basically, the detecting process is a depth-first search. During one vertex visiting, an exhaustive search is applied to detect all of the minimal cycles. As a result, the number of minimal cycles for any vertex can be obtained after the depth-first search. Since the length of a minimal cycle is 4 and the out-degree of each vertex is at most 2, the detection of all of the minimal cycles during one vertex visiting will take  $16 (2^4)$  searches in an exhaustive search. Clearly, the time complexity of an exhaustive search during one vertex visiting is in  $O(1)$  time. In addition, if the adjacent list is applied to store a channel precedence graph, the time complexity of assigning the number of minimal cycles for all vertices will be in  $O(|V| + |A|)$  time. On the other hand, because the out-degree of each vertex is not more than 2, the total number of edges in a channel precedence graph is not more than  $2N$ , where  $N$  is the number of vertices in a channel precedence graph. As mentioned above, by statement (4), there are at least four vertices whose out-degree is not more than 1. Hence, the number of edges in a channel precedence graph is at most  $2N-4$ , in other words,  $O(|A|) = O(|V|) = O(N)$ . Hence, the time complexity of assigning the number of minimal cycles for all vertices is in  $O(N)$  time.

### 3 AN $O(N \log N)$ RDAOA ALGORITHM

In this section, an  $O(N \log N)$  RDAOA algorithm is proposed to minimize the number of switchboxes and assigning the ordering. As mentioned above, the MFVS problem is the most important issue in the RDAOA problem. Therefore, the solution of the MFVS problem for a channel precedence graph will be combined into the RDAOA algorithm.

#### 3.1 RDAOA Algorithm

Due to the properties of a channel precedence graph, the properties will be used as heuristics to develop an efficient algorithm for the minimization of the number of switchboxes and the ordering assignment. By statement (1) and (3) in Lemma 1, the graph is planar and the maximum out-degree of any vertex is 2. Hence,  $(N_{\text{cycle}}, \text{out-degree, in-degree})$  of any vertex can be applied to express the precedence condition for any vertex in the graph. In general, the comparison of  $(N_{\text{cycle}}, \text{out-degree, in-degree})$  is determined by comparing  $N_{\text{cycle}}$ , out-degree and in-degree in a lexicographical order. The RDAOA algorithm is performed as follows: if the in-degree of one vertex in the graph is 0, the vertex will be defined as a channel, assigned an order and deleted from the graph. On the other hand, if the in-degree of no vertex in the graph is 0, the vertex with the largest  $(N_{\text{cycle}}, \text{out-degree, in-degree})$  will be defined as a switchbox and deleted from the graph. Finally, all the defined switchboxes will be assigned the ordering. According to the algorithmic statements, a channel precedence graph is used as the input data and the RDAOA algorithm is described in detail as follows:

**Algorithm RDAOA;**

**Begin**

{Initializationan}

Initialize ORDER = 1, and Compute  $N_{\text{cycle}}$  for all vertices in the graph; Sort the vertices of the graph according to  $(N_{\text{cycle}}, \text{out-degree, in-degree})$ ;

{Define channels and switchboxes and assign the ordering of channels}

**While** (the channel precedence graph is not empty)

**Begin**

**While** (there exists any vertex whose in-degree is 0 in the graph)

**Begin**

Retrieve the vertex and define the vertex as a channel; Assign the order of the channel as ORDER; ORDER = ORDER + 1; Remove the vertex from the graph and change the in-degree of the related vertices;

**End**

Retrieve the vertex with the largest ( $N_{\text{cycle}}$ , out-degree, in-degree) and define the vertex as a switchbox;  
Assign the crossing pins for the switchbox;  
Remove the vertex from the graph and change the in-degree value of the related vertices;

**End**  
{Assign the ordering of switchboxes}  
**While** (any defined switchbox is not assigned the routing order)  
**Begin**  
Remove one defined switchbox without a routing order and assign the order of the defined switchbox as ORDER;  
ORDER = ORDER + 1;  
**End**

### 3.2 Time Complexity Analysis

As mentioned above, for a channel precedence graph, it will take  $O(N)$  time to assign  $N_{\text{cycle}}$  for all vertices. Furthermore, based on ( $N_{\text{cycle}}$ , out-degree, in-degree), the sorting process for all the vertices will take  $O(N \log N)$  time, where  $N$  is the number of vertices in the graph. In general, the vertices can be stored in a priority queue and one vertex can be retrieved according to ( $N_{\text{cycle}}$ , out-degree, in-degree). If the in-degree value of one vertex in the graph is 0, the vertex will be defined as a channel. On the other hand, if not, the vertex with the largest ( $N_{\text{cycle}}$ , out-degree, in-degree) will be defined as a switchbox. Since the maximum out-degree of any vertex is 2, the in-degree change of the related vertices will take  $O(1)$  time after removing one vertex from the graph. Therefore, for the two cases, it will take  $O(\log N)$  time to retrieve one vertex, and the definition of channels and switchboxes will take  $O(N \log N)$  time. Finally, the ordering assignment for the switchboxes will take  $O(N)$  time in the worst case. Hence, the RDAOA algorithm minimizes the number of switchboxes and arranges a safe ordering of channels and switchboxes in  $O(N \log N)$  time, where  $N$  is the number of vertices in the channel precedence graph.

## 4 EXPERIMENTAL RESULTS

The proposed RDAOA algorithm has been implemented using standard C language and run on a SUN workstation under the Berkeley 4.2 UNIX operating system. Many tested examples are applied to measure the number of switchboxes for the RDAOA problem. On the other hand, the three versions, namely first, last, and arbitrary, of the Cai and Wong's algorithm are also implemented and the best result is used to compare the number of switchboxes with the proposed algorithm in Table I.

TABLE I  
The Experimental Results

Example (#V × #H)	Cai and Wong's Algorithm		Our Proposed Algorithm	
	Switchbox	CPU Time	Switchbox	CPU Time
6 × 6	2	0.12 s	2	0.10 s
16 × 18	4	0.31 s	4	0.16 s
23 × 21	6	0.38 s	6	0.24 s
20 × 24	9	0.45 s	9	0.35 s
28 × 35	11	0.82 s	10	0.57 s
37 × 40	15	1.31 s	14	0.89 s
45 × 50	23	2.47 s	21	1.23 s
51 × 49	18	2.36 s	16	1.14 s
71 × 65	25	3.52 s	24	1.63 s

Clearly, the time complexity of the RDAOA problem is reduced from  $O(N^3)$  described in the Cai and Wong's paper to  $O(N \log N)$  in the proposed algorithm. In addition, for the example Ex6, the floorplan graph contains 136 line segments (71 vertical line segments and 65 horizontal line segments) and all the line segments are routed as switchboxes and channels. Finally, 24 switchboxes indicated by thick lines and 112 channels are defined to break all cycles and shown in Figure 4 by the proposed RDAOA algorithm.

## 5 CONCLUSIONS

In a building block placement, the routing space between any pair of adjacent blocks must be defined into a channel or switchbox and assigned the routing order for the routing phase. In this paper, we only introduce linear channels and switchboxes to obtain region definition and ordering assignment. For a channel precedence graph, an  $O(N \log N)$  RDAOA algorithm based on the properties of

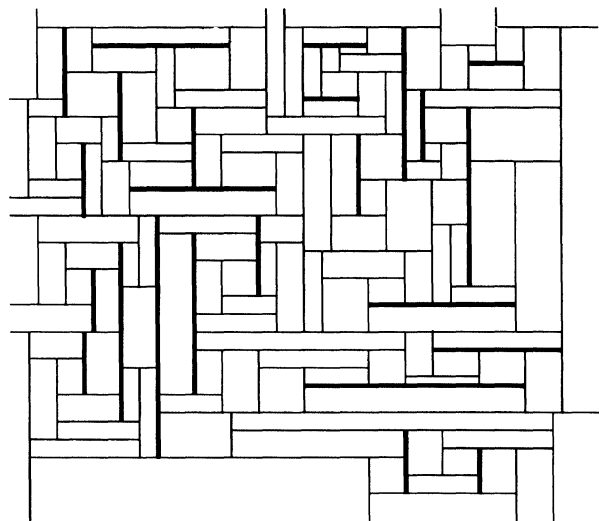


FIGURE 4 An example 71 × 65 (Ex6).

the graph is proposed to minimize the number of switchboxes to increase routability. Several examples have been tested for the proposed algorithm, and the experimental results show that the proposed algorithm has improvement on the number of switchboxes.

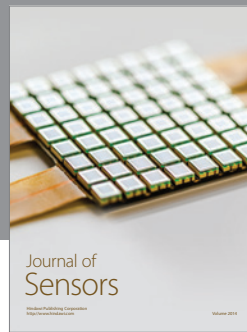
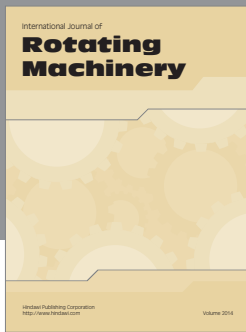
## References

- [1] T. Ohtsuki, Ed., *Advance in CAD for VLSI*, vol. 4, *Layout Design and Verification*. Amsterdam, The Netherlands: North Holland, 1986.
- [2] W. M. Dai, T. Asano and E. S. Kuh, "Routing Region Definition and Ordering Scheme for Building-Block Layout," *IEEE Trans. on Computer Aided Design*, vol. 4, pp. 189–197, 1985.
- [3] R. H. J. M. Otten, "Automatic Floorplan Design." 19th Design Automation Conference, pp. 261–267, 1982.
- [4] T. Chiba, N. Okuda, T. Kambe, I. Nishioka, and S. Kimura, "SHARPS: A Hierarchical Layout System for VLSI," 18th Design Automation Conference, pp. 820–827, 1981.
- [5] S. Kimura, N. Kubo, T. Chiba, and I. Nishioka, "An Automatic Routing Scheme for General Cell LSI," *IEEE Trans. on Computer Aided Design*, vol.2, pp. 285–292, 1983.
- [6] Y. Kajitani, "Order of Channels for Safe Routing and Optimal Compaction of Routing Area," *IEEE Trans. on Computer Aided Design*, vol. 2, pp. 293–300, 1983.
- [7] B. P. Preas and W. M. vanCleemput, "Routing Algorithm for Hierarchical IC Layout," *International Symposium Circuits and Systems*, pp. 482–485, 1979.
- [8] C. P. Hsu, "A New Two-Dimensional Routing Algorithm," 19th Design Automation Conference, pp. 46–50, 1982.
- [9] F. Curatelli, "Region Definition and Ordering for Macrocells with Unconstrained Placement," *INTEGRATION*, the VLSI journal, pp. 169–184, 1991.
- [10] Y. Cai and D. F. Wong, "On Minimizing the Number of L-Shaped Channels in Building-Block Layout," *IEEE Trans. on Computer Aided Design*, vol. 12, pp. 757–769, 1993.
- [11] Y. Cai and D. F. Wong, "Channel/Switchbox Definition Algorithm for VLSI Building-Block Layout," *IEEE Trans. on Computer Aided Design*, vol. 10, pp. 1485–1493, 1991.
- [12] P. Groeneveld, "On Global Wire Ordering for Macro-cell Routing," 26th Design Automation Conference, pp. 155–160, 1989.
- [13] H. Cai, "On Empty Rooms in Floorplan Graphs: comments on a deficiency in two papers," *IEEE Trans. on Computer Aided Design*, vol. 8, pp. 795–797, 1985.
- [14] H. Cai, "Connectivity Biased Channel Construction and Ordering for Building-Block Layout," 25th Design Automation Conference, pp. 560–565, 1988.
- [15] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to The Theory of NP-Completeness*, New York, NY: W.H. Freeman, 1979.

## Biographies

**JIN-TAI YAN** received the B.S., M.S. and Ph.D. degrees in Computer and Information Science from National Chiao Tung University, Taiwan, R. O. C., in 1988, 1990 and 1994, respectively. His main research interests include circuit partitioning, placement and routing in VLSI layout design, high-level synthesis and logic synthesis.

**PEI-YUNG HSIAO** Dr. Hsiao was born on January 5, 1957 in Taiwan, Republic of China. He received the B.S. degree in Chemical Engineering from Tung Hai University in 1980, and the M.S. and Ph.D. degrees in Electrical Engineering from National Taiwan University in 1987 and 1990, respectively. Since August 1990, he has been an Associate Professor in the Department of Computer and Information Science at the National Chiao Tung University, Hsinchu, Taiwan, ROC. From November 1991, he also joins as a Technique Consultant in the Piping Engineering Department, CTCI Corporation, Taipei, Taiwan, ROC. Dr. Hsiao has been a Visiting Senior Fellow in National University of Singapore from July to September, 1993, He ranked 2nd in the 1985 Electronics Engineering Award Examination conducted by the ROC government for studying abroad, and awarded the 1990 Acer Long Term Ph.D. Dissertation Award. His main research interests are VLSI-CAD, Piping Engineering Design Automation, and Neural Network and Expert System Applications and Database System. Dr. Hsiao has published more than 60 articles in conferences and authoritative journals.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

