Region Definition and Ordering Assignment with the Minimization of the Number of Switchboxes

Jin-Tai Yan

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

Abstract----*In this paper, a region definition and ordering assignment (RDAOA) algorithm on minimizing the number of switchboxes is proposed. The time complexity of the algorithm is proved to be in O(n) time, where n is the number of line segments in a given floorplan graph. Finally, several examples have been tested on the proposed algorithm and other published algorithms, and the experimental results show that our algorithm defines fewer switchboxes than other algorithms.*

I. INTRODUCTION

Consider a given building block placement shown in Fig. 1 and suppose that the regions A, B, C, and D are all to be defined as channels. As the terminals in region B are not fixed, 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. The iterative phenomenon of the precedence relations of regions A, B, C and D will construct a channel precedence cycle. The cyclic constraint is named a cyclic channel precedence constraint[1]. Clearly, if a building block placement has some cyclic channel precedence constraints, the routing space will be not defined into only channels and the placement will be a nonslicing placement. For a nonslicing placement, the definition of switchboxes or L-shaped channels is generally introduced to release these cyclic channel constraints and guarantees a safe routing ordering in the routing phase.

For region definition and ordering assignment, the problem of breaking cyclic channel precedence constraints in a building block placement has been extensively studied for many years. Various approaches have been proposed and discussed in previous published papers. First, Otten[2] restricts the acceptable placements of slicing structures in order to avoid these cyclic channel precedence constraints. Hence, it is sure that no cyclic channel precedence constraint is in a building block placement. On the other hand, a building block placement may be modified to avoid these cyclic channel precedence constraints by converting a nonslicing structure into a slicing structure. For example, Chiba[3] perturbs the placement to convert a nonslicing structure into a slicing structure, and Kimura[4] shrinks the placed modules to perform the same conversion. However, instead of breaking the cyclic channel precedence constraints in region definition and ordering assignment, these restricted and modified techniques only change the nonslicing structure into the slicing structure for a building block placement. In recent years, the problem of breaking cyclic channel precedence constraints have been really solved by introducing the definition of switchboxes[5-7] or by combining straight channels into L-shaped channels[1]. In general, for a building block placement, the number of switchboxes is fewer than the number of L-shaped channels. Therefore, the definition of switchboxes is always applied to release all the cyclic channel precedence constraints in a building block placement.

Since the definition of switchboxes releases all the cycles in a channel precedence graph and further yields a safe routing ordering, the routing space will be fully defined into channels and switchboxes. Consider a cyclic channel precedence constraint, we can refer again to the placement shown in Fig. 1. Firstly, the width of region A can be estimated by the position of pins and local connection in region A, and the positions of the floating pins between region A and B are fixed by the crossing distribution of one "T" type junction[8]. Then region B, region C and region D can now be sequentially routed as channels in that order. Finally, region A is routed as a switchbox with a fixed width and fixed terminals on three sides. Hence, the cyclic channel precedence constraint will be released by introducing the definition of a switchbox. For the same reason, all the cyclic channel precedence constraints in a building block placement will be released by introducing the definition of some switchboxes. However, from the viewpoint of detailed routing, the width of a switchbox in region definition is fixed and switchbox routing is not guaranteed. In addition, it is well known that switchbox routing is more difficult than channel routing. Due to the consideration of routing success in the routing phase, it is

important for us to minimize the number of switchboxes in the definition of channels and switchboxes.

To our knowledge, only three algorithms are proposed to minimize the number of switchboxes in region definition and ordering assignment. First, a greedy approach[5] generates the definition of a switchbox whenever a cyclic channel precedence constraint is detected, and always generates the definition of unnecessary switchboxes for a complicated building block placement. Furthermore, Sur-Kolay and Bhattacharya[6] propose a heuristic algorithm based on a clique cover-table approach and a greedy approach to discuss the cycle structure of channel graphs in nonslicible floorplans. In this approach, the definition of unnecessary switchboxes for a complicated building block placement is also generated. Finally, Cai and Wong[7] propose a graph-based algorithm to find a feedback vertex set in a channel precedence graph by the transformation between a reduced digraph and an intersection graph, and obtain significant improvement on the number of switchboxes. However, the time complexity of the algorithm is in $O(n^3)$ time.

II. PRELIMINARIES

Basically, a floorplan graph only consists of horizontal and vertical line segments. In general, each horizontal or vertical line segment in a floorplan graph will be represented as one channel. As two rows of terminals located on a channel are fixed, the channel will be routed successfully by a channel router. Hence, for one "T" type junction in a floorplan graph, the base channel of the "T" type junction must be routed before the top channel of that according to the specification of channel routing. Based on the precedence relation in one "T" type junction, we define a channel precedence graph from a floorplan graph as follows :

Definition 1 : For a given 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 one vertex of V. Each directed edge (u, v) is in A if and only if the two line segments corresponding to u and v form one "T" type junction in the floorplan graph, and the one corresponding to u is the base of the "T" type junction.

In Fig. 2(a), the floorplan graph for a building block placement is shown, where vi denotes the i-th vertical line segment from the left, and the hj denotes the j-th horizontal line segment from the top. The channel precedence graph G of Fig. 2(a) is illustrated in Fig. 2(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) }.

Due to the geometrical properties in a floorplan graph, the generated channel precedence graph has several available graphical properties to help solve the RDAOA problem of minimizing the number of switchboxes.

Lemma 1 : Let G(V, A) be a channel precedence graph, 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 number of edges |A| is at most 2n - 4, where n is the number of vertices.

From the viewpoint of a channel precedence graph, if a channel precedence graph is cyclic, a set of vertices will be removed from the graph to generate a new acyclic channel precedence graph. Clearly, the set of removed vertices in a channel precedence graph will correspond to the definition of switchboxes in a mapped building block placement. As the set of removed vertices is defined as switchboxes, all the cyclic channel precedence constraints in the mapped building block placement will be fully released by the definition of switchboxes. Furthermore, the vertices in the remaining acyclic channel precedence graph will be only defined as channels. Finally, a safe routing ordering of these channels will be assigned by a topological sorting, and the ordering of the defined switchboxes will be further assigned randomly after all of the defined channels.

Therefore, the RDAOA problem of minimizing the number of switchboxes in a building block placement will correspond to the problem of minimizing the number of the set of removed vertices in a generated channel precedence graph. For a channel precedence graph, the problem of minimizing the number of the set of removed vertices is defined as the MFVS problem.

Definition 2 : A feedback vertex set S of a directed graph G = (V, A) is a subset of V. The vertex removal from G results in an acyclic directed graph. A minimum feedback vertex set S_{min} is a feedback vertex set with minimum cardinality.

It is well known that the minimum feedback vertex set (MFVS) problem[9] remains NP-hard for a channel precedence graph. Since the MFVS problem is to remove a minimal set of vertices to break all the cycles in a channel precedence graph, it will be important for the development of a heuristic algorithm to study the characterization of a channel precedence cycle in a channel precedence graph. According to the construction of a floorplan graph, some graphical properties of a channel precedence cycle are further described.

Lemma 2 : For a channel precedence graph, three graphical properties of a channel precedence cycle are as follows :

(1). The length is even.
(2). The length of a minimal cycle is 4.

(3). All vertices are in at most four minimal cycles at the same time.

In general, the cycles in a channel precedence graph are divided into the minimal cycles whose length is 4 and the long cycles whose length is more than 4. According to the physical structure of a floorplan graph, it may be assumed that most of the cycles in a channel precedence graph are minimal cycles. Therefore, it is important for the MFVS problem to break all the minimal cycles in a channel precedence graph. The MFVS problem in a channel precedence graph will be solved by the minimal-cycle phase and the long-cycle phase[6]. In the minimal-cycle phase, all of the minimal cycles and most of long cycles in the channel precedence graph are broken by removing a minimal of vertices to be defined as switchboxes. Finally, the remaining long cycles are broken by an iterative greedy approach in the long-cycle phase.

Since the minimal-cycle phase is important for the MFVS problem, some available graphs need to be further defined for the minimal-cycle phase. First, a weighted channel precedence graph is established to indicate cyclic information by assigning weight onto the vertices in a channel precedence graph.

Definition 3 : A weighted channel precedence graph G' = (V', A') is a channel precedence graph with weight assigned to V'. For each vertex u in V', a weight value w(u) is assigned onto the vertex u, where w(u) is the number of the minimal cycles in which u is contained.

In order to define a set of vertices in a channel precedence graph as switchboxes in the minimal-cycle phase, some vertices in a weighted channel precedence graph are of no use. Furthermore, a cyclic channel precedence graph is further established to simplify the size of the weight channel precedence graph by removing the vertices which are not in any minimal cycle.

Definition 4 : A cyclic channel precedence graph G* = (V*, A*) is a subgraph of a weighted channel precedence graph by removing the vertices which are not in any minimal cycle, where V* is a subset of V, and A* is a subset of A.

Fig. 3(a) shows a weighted channel precedence graph of Fig. 2(b), and a cyclic channel precedence graph of Fig. 3(a) is shown in Fig. 3(b). In Fig. 3(b), four minimal cycles $\{ h_1, v_2, h_2, v_1 \}$ $\{ h_2, v_5, h_3, v_3 \}$, $\{ h_3, v_4, h_4, v_3 \}$ and $\{ h_3, v_6, h_5, v_4 \}$ are in the cyclic channel precedence graph.

Furthermore, by Lemma 2, all vertices in a channel precedence graph are in at most four minimal cycles at the same time. For the same reason, all vertices in a cyclic channel precedence graph are in at least one minimal cycle and in at most four minimal cycles at the same time. Therefore, one vertex in a cyclic channel precedence graph will be contained in one minimal cycle, two minimal cycles, three minimal cycles or four minimal cycles simultaneously. A k-cycle pattern in a cyclic channel precedence graph will be defined to optimally find a minimal set of vertices to be defined as switchboxes in the minimal-cycle phase.

Definition 5 : A k-cycle pattern is a connected subgraph of a cyclic channel precedence graph and contains k connected minimal cycles. The weight of at least one common vertex in the k-cycle pattern is exactly k.

Clearly, only 1-cycle, 2-cycle, 3-cycle and 4-cycle patterns will be found in a cyclic channel precedence graph. All of the possible k-cycle patterns are shown in Fig. 4. For the cyclic channel precedence graph in Fig. 3(b), it covers one 2-cycle pattern and one 3-cycle pattern. The 2-cycle pattern is formed by two minimal cycles $\{ h_1, v_2, h_2, v_1 \}$ and $\{ h_2, v_5, h_3, v_3 \}$. The other 3-cycle pattern is formed by three minimal cycles $\{ h_2, v_5, h_3, v_3 \}$, $\{ h_3, v_4, h_4, v_3 \}$ and $\{ h_3, v_6, h_5, v_4 \}$.

According to the definition of a k-cycle pattern, all of the minimal cycles in a k-cycle pattern will be broken by removing one common vertex. Hence, a k-cycle pattern will be viewed as a processed unit in the minimal-cycle phase. For any pair of k-cycle patterns in a cyclic channel precedence graph, the intersection relation will be set by sharing at least one minimal cycle. Here, a k-cycle intersection graph is further established by modeling the

intersection relation between any pair of adjacent k-cycle patterns.

Definition 6 : For a cyclic channel precedence graph, a k-cycle intersection graph $G_{k\text{-}cycle} = (V_{k\text{-}cycle}, E_{k\text{-}cycle})$ is an undirected graph, where $V_{k\text{-}cycle}$ represents all of the k-cycle patterns in the cyclic channel precedence graph, and $E_{k\text{-}cycle}$ represents the intersection relations of all pairs of adjacent k-cycle patterns.

Lemma 3 : For a floorplan graph, the number of defined switchboxes in the minimal-cycle phase is at most the number of vertices in its k-cycle intersection graph.

For a k-cycle intersection graph, each vertex represents a k-cycle pattern. Hence, the vertex set is further divided into the independent set and the dependent set. For an independent vertex, the mapped k-cycle pattern will be independently broken in the minimal-cycle phase. On the other hand, for a dependent vertex, the mapped k-cycle pattern will not be independently broken in the minimal-cycle phase. Here, the definition of an independent vertex in a k-cycle intersection graph will be applied to define necessary switchboxes for its related building block placement. Furthermore, according to the construction of a k-cycle intersection graph, some graphical properties are further described.

Definition 7 : For one vertex in a k-cycle intersection graph, the vertex represents a k-cycle pattern. If there exists any minimal cycle only in the k-cycle pattern, the vertex will be defined as an independent vertex. On the other hand, if all the minimal cycles in the k-cycle pattern are fully covered by the other k-cycle patterns, the vertex will be defined as a dependent vertex.

Lemma 4 : For a k-cycle intersection graph, some graphical properties are as follows :

(1) Planar.
(2) The intersection of two k-cycle patterns has at most two minimal cycles.
(3) The degree of at least one vertex is less than 3.
(4) One k-cycle pattern is not fully covered in the other k-cycle pattern.
(5) Any vertex with degree 1 is an independent vertex.

In Fig. 5(a), the floorplan graph of a building block placement is illustrated and all the k-cycle patterns are labeled by circles. Furthermore, Fig. 5(b) shows and explains the related k-cycle intersection graph for the floorplan graph. For the k-cycle intersection graph, it is clear that vertex 1, 3 and 6 are independent vertices and the other vertices are dependent vertices.

III. REGION DEFINITION AND ORDERING ASSIGNMENT

As mentioned above, the RDAOA problem of minimizing the number of switchboxes in a building block placement will correspond to the MFVS problem for a channel precedence graph. Based on the solution of the MFVS problem for a channel precedence graph, an efficient RDAOA algorithm for minimizing the number of switchboxes is proposed. Basically, the solution of the MFVS problem for a channel precedence graph is obtained by the minimal-cycle phase and the long-cycle phase.

In the minimal-cycle phase, a channel precedence graph will be transformed into a k-cycle intersection graph in O(n) time, where n is the number of vertices in a channel precedence graph. In the k-cycle intersection graph, each vertex represents one k-cycle pattern. As one vertex in the k-cycle intersection graph is selected, all the minimal cycles in the related k-cycle pattern will be broken by removing one common vertex to be defined as a switchbox. Hence, by breaking these selected the k-cycle patterns, all the minimal cycles in a channel precedence graph will be broken by removing a minimal set of vertices to be defined as switchboxes.

However, based on the physical structure of a floorplan graph, it is clear that some long cycles share vertices with minimal cycles. Furthermore, it is desirable that most of the long cycles can be broken at the same time in the minimal-cycle phase. Hence, in addition to the vertex weight, the second heuristic in-degree will be introduced to break one selected k-cycle pattern, that is, if the selected k-cycle pattern has at least two common vertices, one common vertex with larger in-degree will be removed to break the k-cycle pattern. As a result, all the minimal cycles and most of the long cycles in a channel precedence graph will be broken at the same time in the minimal-cycle phase, and the minimal-cycle phase will be solved by the following Minimal_Cycle algorithm.

In the algorithm, the procedure of removing a minimal set of vertices to break all the minimal cycles and most of long cycles is not stopped until the k-cycle intersection graph is empty. Basically, the operations in a loop statement are divided into four main steps : (1) Vertex Selection, (2) Vertex Removal, (3) Vertex Deletion and (4) Vertex Search. According to the description of the main steps, the Minimal_Cycle algorithm will be shown in Fig. 6.

```
Algorithm Minimal_Cycle
Input : a channel precedence graph;
    Begin
        Call the CPG-CCPG algorithm;
        Call the CPG-KIG algorithm;
        While ( KIG is not empty )
            Begin
                { Vertex Selection }
                If ( there exists any independent vertex u in KIG )
                    Retrieve the vertex u ;
                Else
                    Retrieve any vertex u with degree 2;
                Endif
                { Vertex Removal }
                Select the vertex v with maximal (vertex-weight, in-degree) in
                    the related k-cycle pattern P(u);
                Mark the broken minimal cycles in the cyclic channel
                    precedence graph;

                S_min = S_min ∪ { v };
                { Vertex Deletion }
                Modify the k-cycle intersection graph by deleting the vertices
                    which connect the vertex u and whose unbroken
                    minimal cycles are fully covered by one adjacent
                    k-cycle pattern;
                { Vertex Search }
                Find new independent vertices from adjacent dependent
                    vertices;
            End
        Return the vertex set, S_min;
    End
```

Fig. 6 The Minimal_Cycle algorithm.

On the other hand, the remaining long cycles in the channel precedence graph will be broken by an iterative channel/switchbox definition and ordering assignment in the long-cycle phase. If there exists one vertex with in-degree 0, the vertex will be removed from the channel precedence graph, defined as a channel and assigned a routing order. If there exists one vertex with out-degree 0, the vertex will be removed from the channel precedence graph, defined as a channel and pushed into a stack. Based on the physical structure of a floorplan graph, the remaining long cycles may share vertices each other. In order to minimize the number of switchboxes in the long-cycle phase, the first heuristic out-degree and the second heuristic in-degree are applied to break the remaining long cycles. Until there exists no vertex with in-degree 0 or out-degree 0, one vertex with the largest (out-degree, in-degree) will be removed from the channel precedence graph and defined as a switchbox. As the channel precedence graph is empty, the channels in the stack will be popped and assigned a routing order. Finally, all the vertices defined as switchboxes will be randomly assigned routing orders. According to the description of the algorithm, the RDAOA algorithm is shown in Fig. 7.

```
Algorithm RDAOA
Input : a floorplan graph;
Begin
    ORDER = 1; S_min = ∅ ; Stack = ∅;
    Call the FG-CPG algorithm;
    { The minimal-cycle phase }
    Call the Minimal_Cycle algorithm and Return a minimal set of vertex for the
        minimal cycles, S_min;
    Define all of the vertex in S_min as switchboxes;
    Assign the crossing pins for the switchboxes;
    Construct a new channel precedence graph by deleting S_min from the original
        channel precedence graph;
    { The long-cycle phase }
    While ( the channel precedence graph is not empty )
        Begin
            While (there exists any vertex w with in-degree 0 in the channel
                precedence graph)
                Begin
                    Define the vertex w as a channel;
                    Assign the order of the channel as ORDER;
                    ORDER = ORDER +1;
                    Remove the vertex w from the graph, and change the in-degree value of
                        the related vertices;
                End
            While (there exists any vertex w with out-degree 0 in the channel
                precedence graph)
                Begin
                    Define the vertex w as a channel;
                    Push the vertex w into Stack;
                    Remove the vertex w from the graph, and change the out-degree value
                        of the related vertices;
                End
            Retrieve the vertex w with the largest (out-degree, in-degree);
            Define the vertex w as a switchbox;

            S_min = S_min ∪ { w };
            Assign the crossing pins for the switchbox;
            Remove the vertex w from the graph, and change the in-degree value of the
                related vertices;
        End
    While ( Stack is not empty )
        Begin
            Pop one vertex w in Stack;
            Assign the order of the channel as ORDER;
            ORDER = ORDER + 1;
        End
    While ( S_min is not empty )
        Begin
            Retrieve any vertex w in S_min;
            Assign the order of the switchbox as ORDER;
            ORDER = ORDER + 1;
        End
    End
```

Fig. 7 The RDAOA algorithm.

Theorem 1 **:** The time complexity of the RDAOA algorithm is in O(n) time, where n is the number of line segments in a floorplan graph.

IV. EXPERIMENTAL RESULTS

Our proposed algorithm has been implemented using standard C language and run on a SUN workstation under the Berkeley 4.2 UNIX operating system. Thirty examples including example Ex6 in the Cai and Wong's paper[7] are applied to measure the number of switchboxes in the RDAOA problem. On the other hand, an exhaustive algorithm based on the detection of all the cycles in a channel precedence graph and the exhaustive search of the clique cover problem is implemented to obtain the optimal solution of the number of switchboxes for these tested examples. Based on the optimal results of the exhaustive algorithm, a greedy algorithm[5] and the three versions, namely first, last, and arbitrary, of the Cai and Wong's algorithm[7] are also implemented to compare the number of switchboxes in the RDAOA problem. In the experimental results, the results of the greedy algorithm and the best result on these three versions of the Cai and Wong's algorithm are applied to compare the number of switchboxes with the results of our proposed algorithm.

In Table I, the experimental results of twelve representative examples with different sizes will be shown and compared on the exhaustive algorithm, the greedy algorithm, the Cai and Wong's algorithm and our proposed algorithm. From the viewpoint of the numerical results, the contribution of our proposed algorithm is not only in the theoretical reduction of time complexity but also in the number of switchboxes for the tested examples.

First, as mentioned above, the time complexity of our proposed algorithm in the RDAOA problem of minimizing the number of switchboxes is reduced from $O(n^3)$ described in the Cai and Wong's paper to O(n) time. Second, an

optimal ratio $\gamma_{optimal}$ is further applied to measure optimal degree of a proposed algorithm for any tested example such as

$$\gamma_{optimal}(Alg, Ex) = \frac{N_{opt}(Ex)}{N(Alg, Ex)} \times 100\%,$$

where $N_{optimal}(Ex)$ is the number of switchboxes by an exhaustive algorithm for the example Ex, and N(Alg, Ex) is the number of switchboxes by a proposed algorithm Alg for the example Ex. The definition of the optimal ratio is applied to the greedy algorithm the Cai and Wong's algorithm and our proposed algorithm for all the tested examples. It is clear that our proposed RDAOA algorithm attains 100% optimal ratio for all the tested examples. The fact proves that the proposed algorithm is efficient for minimizing the number of switchboxes in the RDAOA problem.

For example 39x43, it contains 39 vertical line segments and 43 horizontal line segments in a floorplan graph, and these line segments will construct 37 minimal cycles and 9 long cycles in the floorplan graph. As a result, 37 minimal cycles and 9 long cycles will be broken in the minimal-cycle phase and 0 long cycle will be broken in the long-cycle phase. By running our algorithm, 14 switchboxes are defined to break all of the minimal cycles for the RDAOA problem and the experimental result is shown in Fig. 8. Furthermore, for example Ex6 in the Cai and Wong's paper[7], the experimental result is obtained by our algorithm and shown in Fig. 9. The floorplan graph of example Ex6 contains 71 vertical line segments and 65 horizontal line segments. By the detection of an exhaustive search, the floorplan graph contains 35 minimal cycles and 12 long cycles. As a result, 24 switchboxes indicated by thick lines and 112 channels are defined to break all the minimal cycles and the long cycles.

REFERENCES

[1] 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, pp189-197, 1985.

[2] R. H. J. M. Otten, "Automatic Floorplan Design" Proc. 19th Design Automation Conference, pp261-267, 1982.

[3] T. Chiba, N. Okuda, T. Kambe, I. Nishioka, and S. Kimura, "SHARPS : A Hierarchical Layout System for VLSI" in Proc. 18th Design Automation Conference, pp. 820-827,1981.

[4] 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.

[5] B. P. Preas and W. M. vanCleemput, "Routing Algorithm for Hierarchical IC Layout," Proc. International Symposium Circuits and Systems, pp. 482-485, 1979.

[6] S. Sur-Kolay and B. B. Bhattacharya, "The Cycle Structure of Channel Graphs in Nonslicible Floorplans and A Unified Algorithm for Feasible Routing Order," International Conference on Computer Design, pp. 524-527, 1991.

[7] Y. Cai and D. F. Wong, "Channel/Switchbox Definition for VLSI Building-Block Layout," IEEE Trans. on Computer Aided Design, vol. 10, pp. 1485-1493, 1991.

[8] J. T. Yan and P. Y. Hsiao, "Routability Crossing Distribution and Floating Terminal Assignment for T-type Junction Region," Fourth Great Lakes Symposium on VLSI, pp.162-165, 1994.

[9] 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.

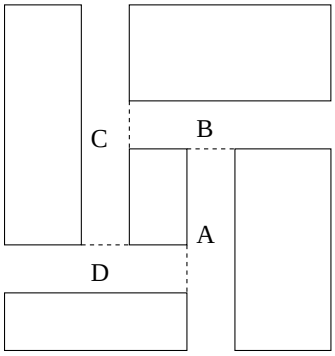


Fig. 1 Cyclic channel precedence constraint.

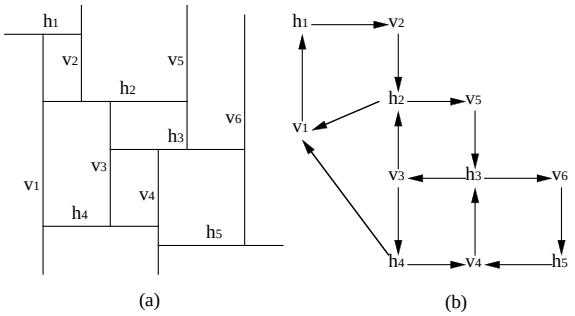


Fig. 2 (a) A given floorplan graph.
(b) A channel precedence graph.

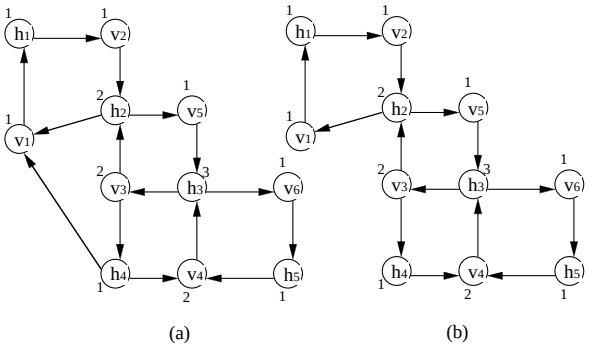


Fig. 3 (a) A weighted channel precedence graph.
(b) A cyclic channel precedence graph.

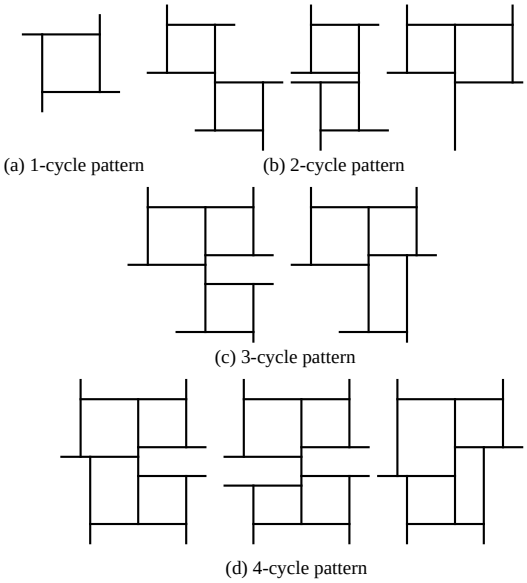
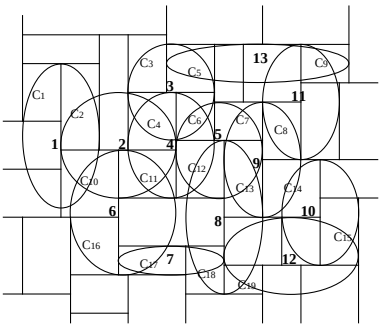


Fig.4 All possible k-cycle patterns.



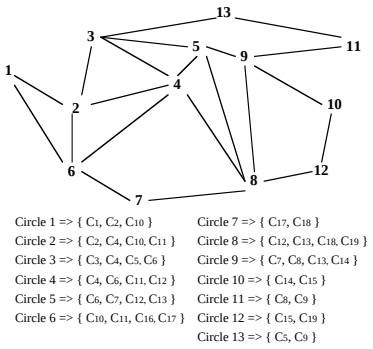


Fig. 5 (a) A given floorplan graph.
(b) A k-cycle intersection graph.

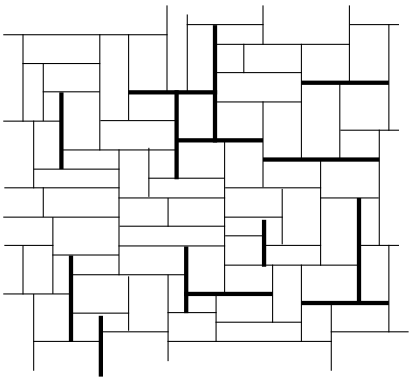


Fig. 8 An example 39 x 43

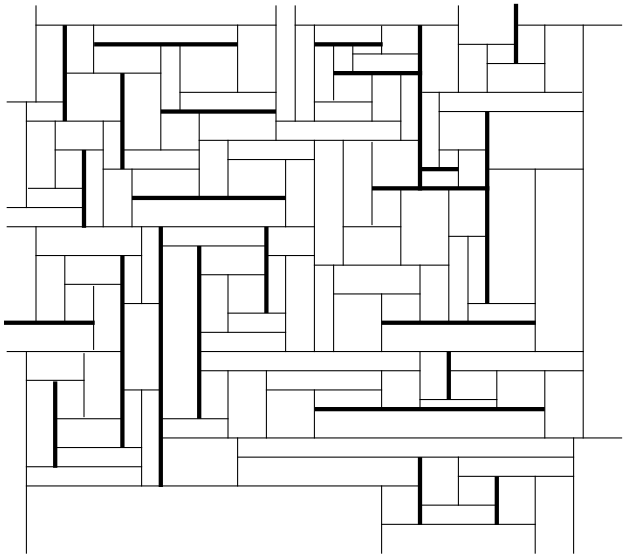


Fig.9 An example 71 x 65 (Ex6)

Table I The Experimental Results

Example (#V x #H)	# minimal cycle	# long cycle	Exhaustive Search		Greedy Algorithm			Cai and Wong's Algorithm			Our RDAOA Algorithm		
			#S	Time	#S	Time	$\gamma_{optimal}$	#S	Time	$\gamma_{optimal}$	#S	Time	$\gamma_{optimal}$
6 x 6	3	0	2	0.3 s	5	0.1 s	40.0%	2	0.1 s	100%	2	0.1 s	100%
13 x 12	9	1	4	1.5m	8	0.2 s	50.0%	4	0.2 s	100%	4	0.2 s	100%
16 x 18	10	0	4	6.1m	11	0.3 s	36.4%	4	0.5 s	100%	4	0.4 s	100%
23 x 21	11	1	6	14.2m	15	0.6 s	40.0%	6	0.8 s	100%	6	0.7 s	100%
20 x 24	16	3	9	31.3m	20	0.9 s	45.0%	9	1.2 s	100%	9	1.0 s	100%
28 x 35	21	4	10	45.6m	24	1.2 s	41.7%	10	1.6 s	100%	10	1.4 s	100%
32 x 42	31	6	15	1.2 h	33	1.4 s	45.5%	16	1.8 s	93.8%	15	1.7 s	100%
37 x 40	38	7	14	2.1 h	32	1.7 s	43.8%	15	2.3 s	93.3%	14	2.0 s	100%
39 x 43	37	9	14	1.7 h	34	1.8 s	41.2%	14	2.7 s	100%	14	2.2 s	100%
45 x 50	46	10	21	3.7 h	45	2.4 s	46.7%	22	3.1 s	95.5%	21	2.6 s	100%
51 x 49	54	8	16	5.2 h	41	2.6 s	39.0%	17	3.5 s	94.1%	16	2.8 s	100%
71 x 65	35	12	24	8.5 h	49	3.0 s	49.0%	25	4.1 s	96.0%	24	3.2 s	100%