직선으로 둘러싸인 영역과 비평면적 표면 상에서의 회로 분할과 배치를 위한 그래프 매칭 알고리즘

박 인 철°, 경 종 민
한국과학기술원, 전기 및 전자공학과

A Graph Matching Algorithm for Circuit Partitioning and Placement in Rectilinear Region and Nonplanar Surface

In-Cheol Park, Chong-Min Kyung
Department of Electrical Engineering, KAIST

Abstract

This paper proposes a graph matching algorithm based on simulated annealing, which assures the globally optimal solution for circuit partitioning for the placement in the rectilinear region occurring as a result of the pre-placement of some macro cells, or onto the nonplanar surface in some military or space applications. The circuit graph (G_C) denoting the circuit topology is formed by a hierarchical bottom-up clustering of cells, while another graph called region graph (G_R) represents the geometry of a planar rectilinear region or a nonplanar surface for circuit placement. Finding the optimal many-to-one vertex mapping function from G_C to G_R, such that the total mismatch cost between two graphs is minimal, is a combinatorial optimization problem which was solved in this work for various examples using simulated annealing.

I. Introduction

VLSI layout problem is how to place each block of a given circuit on a given region such that the given cost function such as overall routing length is minimized. Although a significant advancement has been made in the VLSI algorithm area for the last decade, most of the earlier works are dealing with the placement of one type of cell in a rectangular region. In practical VLSI layout, however, the region available for circuit placement is rectilinear, not necessarily rectangular, as a result of pre-placement of some macro blocks as shown by a typical VLSI chip floorplan in Fig.1. Chi[1] reported an algorithm for embedding a given circuit within a rectilinear region based on successive min-cut bi-partitioning of both regions and circuit, respectively. However, owing to the greedy nature, it cannot assure the globally optimal distribution of circuit modules and nets over the whole rectilinear region.

In this paper, we describe a new approach called "graph matching"[2] which assure globally optimal result in the partitioning of a given eletronic circuit into each rectangular subregion of the whole rectilinear region. The overall procedure for circuit palcement in rectilinear region according to this scenario consists of four steps as shown in Fig.2. The procedure starts with i) step 1; converting the given rectilinear region into a graph called region graph (G_R) and ii) step 2; converting the given circuit into a graph called circuit graph (G_C). Step 1 of dividing a rectilinear region into rectangles can be done using various heuristics[1,3] depending on the situation. Step 2 of forming the circuit graph, G_C is responsible for reducing the number of vertices through a bottom-up clustering explained in section II. Given a region graph (G_R) and a circuit graph (G_C), the step 3 finds the globally optimal assignment of

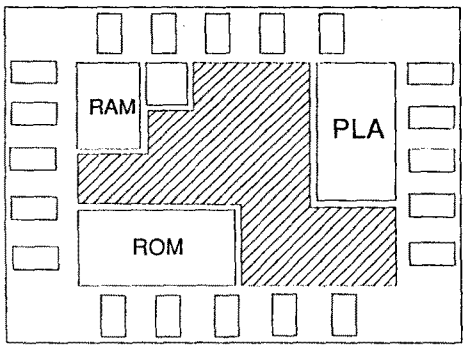


Fig. 1 A typical floorplan of VLSI chip.

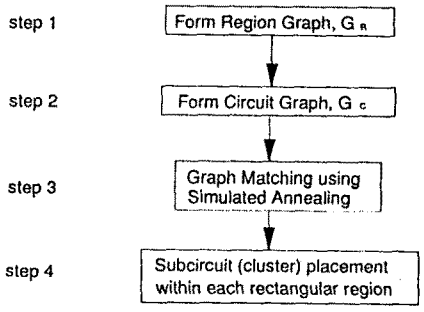


Fig. 2 A procedure for placement in rectilinear region.

vertices in G_C onto each vertex of G_R, using the graph matching procedure based on simulated annealing[4] mentioned in section III. After the circuit partitioning, the placement of each subcircuit within each rectangular region can be done using various existing algorithms[5,6].

II. Circuit Graph Formation Using Bottom-Up Clustering

Circuit graph G_C is a graph whose vertices represent the circuit modules, while the edges represent the inter-module connections. Bottom-up clustering of strongly interconnected cells is necessary in building the circuit graph because reducing the number of vertices in G_C, which is usually much larger than the number of vertices in G_R, through clustering is very important to reduce the complexity of the graph matching problem.

Fig. 3 shows two clusters as a result of successive bottom-up clustering. Following algorithm is a procedure for clustering such that the total number of clusters is less than a constant (CPR) times the num_region, number of vertices in G_R, while the size of each cluster is limited by max_size.

```
BU_Clustering
{
    while( |V_c| < num_region * CPR ) {
        G_c=(V_c,E_c) ;      /* construct G_c, circuit graph */
        avg := average of edge weight, w(e), e ∈ E_c;
        for (e=(v_i,v_j) in descending order of w(e) ) {
            if( w(e) < avg ) break ;
            else if( w(v_i) + w(v_j) < max_size) {
                /* w(v_i) denotes the weight of vertex i, or
                   total area of cells within cluster i */
                merge ( v_i,v_j ) ;
            }
        }
        max_size := β * max_size ;
    }
}
```

Algorithm 1 : Hierarchical Bottom-Up Clustering

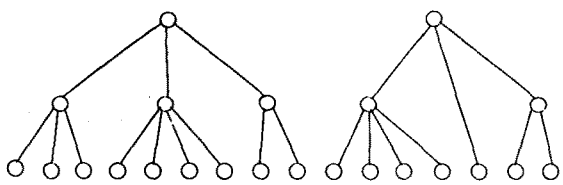


Fig. 3 Cluster forest

III. Graph Matching

Fig. 4(a) shows an example of rectilinear region divided into rectangular subregions, and its corresponding region graph, G_R is shown in Fig. 4(b), where the weight of each vertex denotes the area of the corresponding rectangular subregion, while the edge weight denotes the length of the border line between two neighboring rectangular subregions. Fig. 4(c) shows that a circuit graph (G_C) having 10 vertices (denoted as small circles) is embedded onto the region graph (G_R) having 4 vertices (denoted as large circles) through an assignment of vertices V_C to V_R. (V_C and V_R denote the vertex sets of G_C and G_R, respectively.) The graph matching problem is formally stated here as follows :

> Given two graphs, $G_R = (V_R, E_R)$ and $G_C = (V_C, E_C)$, find the mapping from V_C to V_R such that the given global cost is minimal.

The number of vertices in G_C is larger than the number of vertices in G_R (even after the bottom-up clustering), i,e., $|V_C| > |V_R|$, and the graph matching problem is to evaluate the cost function for each possible many-to-one vertex mapping from V_C to V_R. Simulated annealing has been used in our work to handle this combinatorial optimization problem.

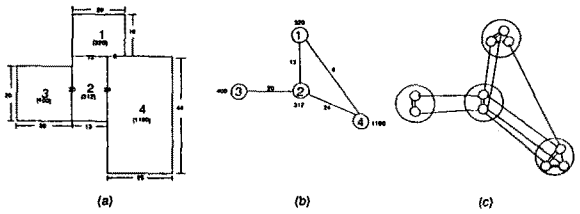


Fig. 4 (a) Subdivision of a rectilinear region
(b) Corresponding region graph
(c) Circuit graph

Three kinds of cost terms were incorporated; half-perimeter cost, edge mismatch cost and vertex mismatch cost. The half-perimeter cost of a net is the half-perimeter length of the MBB (Minimum Bounding Box) enclosing all the center points of the rectangular subregions on which at least one circuit module belonging to the net was assigned. The half-perimeter cost of a net having its circuit modules assigned onto the rectangular subregions, 1,2 and 4 in Fig. 5(a), is the half-perimeter of the MBB shown as a dotted line. The second term is the vertex mismatch cost which represents the difference between the area of a rectangular subregion and the sum of areas of the circuit modules assigned to that rectangular subregion. Finally, the edge mismatch cost represents the difference between the capacity of a border line and the sum of weights of all nets passing through that border line.

Determining the actual border line segments (corresponding to the edges in G_R) being crossed over by each net is at least as complicated as the global routing problem which is NP-complete. For simplicity, It is assumed here that all the clusters assigned to a rectangular region are placed at the center of that rectangle, and only those edges (denoted as thick lines in Fig. 5(b)) corresponding to the border line segments fully or partially enclosed by the

MBB of a net (denoted as dotted rectangle in Fig. 5(a)) are responsible for providing the channel for routing the net. The total cost function used in our simulated annealing is a linear superposition of the three cost terms mentioned.

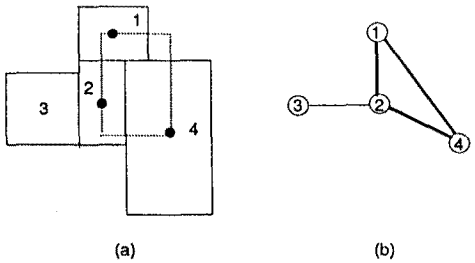


Fig. 5 (a) A net consisting of three modules.
(b) Region graph where the thick edges denote boundary segments consumed by the net.

IV. Circuit Placement on Nonplanar Surface

Another possible application of the proposed graph matching is the circuit placement on nonplanar surfaces. Although only planar surfaces are considered for VLSI layout, there are some applications where the electronic control circuitry should be placed or "pasted" on the interior or exterior surface of 3-dimensional objects, for example, in military or space applications.

Fig. 6(a) shows an orthogonal hexahedron (3-D box) for nonplanar surface embedding of electronic circuits. The six faces of the 3-D box are shown in (b), while the corresponding region graph is shown in (c). The assignment of circuit modules onto each face is the similar graph matching problem. (The region graph in (c) is planar, however.) Fig.7 shows how to determine the border line segments being crossed by a net according to the various distribution configuration, from (a) to (i), of the circuit modules on each face of the 3-D box. Only those border line segments partially or fully enclosed with the MBV(Minimum Bounding Volume) of a net are used for routing the net. For example, Fig. 7(a) represents the case when all the circuit modules of a net are assigned onto one face, in which case no border line segment is crossed by the net.

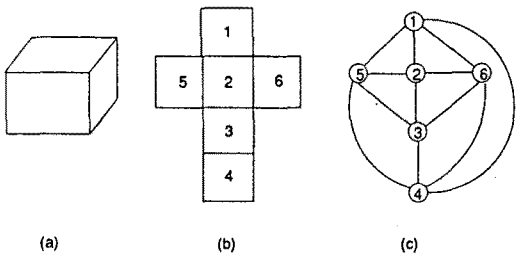


Fig. 6 (a) Othogonal hexahedron
(b) Six faces of the hexahedron
(c) Corresponding region graph

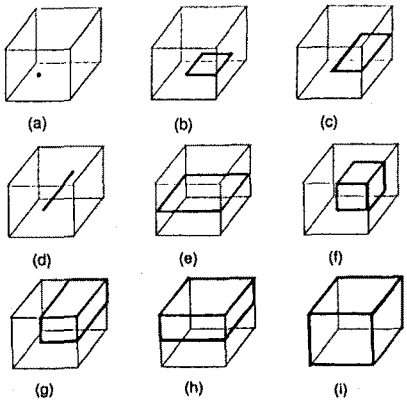


Fig. 7 Minimum bounding volumes

V. Experimental Results and Conclusions

Fig. 8 shows, in a spanning tree style, the connection pattern of a circuit having 67 modules after the assignment onto each rectangular subregion using the graph matching procedure based on simulated annealing. The size and the location of each circuit module within each rectangular subregion is meaningless and only for the purpose of illustration. However, it can be readily seen that the number of vertices assigned onto a subregion is quite proportional to the area of the subregion, which means the vertex mismatch cost is small.

Fig. 9(a) and (b) show additional rectilinear regions used for the placement of two benchmark circuits[7] whose statistical data are shown in Table 1. Table 2 and 3 show the detailed results of the graph matching based on simulated annealing, where the vertex mismatch and edge mismatch costs are shown for each vertex and edge of the region graph. Finally, table 4 shows the similar result for the assignment of a circuit having 144 modules onto 6 faces of a 3-D box using the graph matching algorithm.

In conclusion, a graph matching algorithm for finding the globally optimal circuit partitioning and placement on the rectilinear region occurring due to the pre-placement of macro cells and on the nonplanar surface is proposed based on simulated annealing.

References

[1] M.C. Chi, "An automatic rectilinear partitioning procedure for standard cells", 24-th DAC, 1987

[2] C.C. Shen, W.H. Tsai, "A graph matching Approach to optimal task assignment in distributed computing systems using a minimax criterion", IEEE Trans. on Computers, Vol. c-34, No. 3, March, 1985

[3] T. Ohtsuki, "Minimum dissection of rectilinear region", ISCAS, pp 1210-1213, 1982

[4] S. Kirkpatrick, C.D. Gelatt, Jr., M.P.Vecchi, "Optimization by simulated annealing", Science, Vol. 220, 1983

[5] A.E. Dunlop, B.W. Kernighan, "A procedure for layout of standard cell VLSI circuits", IEEE Trans. on CAD, Vol. CAD-4, No. 1, 1985

[6] C. Sechen and A. Sangiovanni-Vincentelli, "The TimberWolf placement and routing package", IEEE Journal of solid state circuits, Vol. sc-20, No. 2, 1985

[7] B. Preas, "Bechmarks for cell-based layout systems", 24-th DAC, 1987

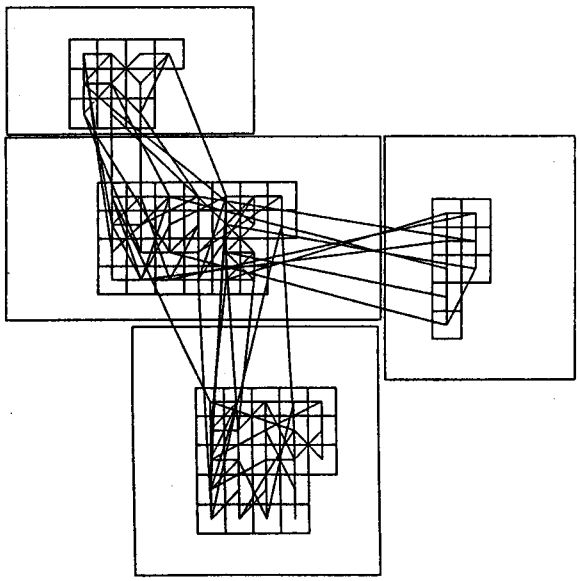
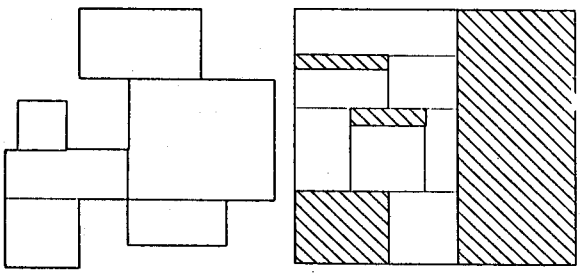


Fig. 8 Spanning tree connection pattern of a circuit having 67 modules.



(a) (b)

Fig. 9 Region pattern for (a) Primary 1, (b) Test 4.

	Vertex weight in G_R $W_R(v_i)$	Vertex weight in G_c $W_c(v_i)$	mismatch cost $\lvert W_R(v_i)-W_c(v_i)\rvert$	
v_1	15825	15850	25	(0.16 %)
v_2	4220	4140	80	(1.89 %)
v_3	10550	10450	100	(0.95 %)
v_4	9495	9490	5	(0.05 %)
v_5	31650	31800	150	(0.47 %)
v_6	8440	8450	10	(0.11 %)
Total	80180	80180	370	(0.46 %)
	Edge weight in G_R $W_R(e_i)$	Edge weight in G_c $W_c(e_i)$	mismatch cost $\lvert W_R(e_i)-W_c(e_i)\rvert$	
e_1	134	137	3	(2.23 %)
e_2	102	101	1	(0.99 %)
e_3	68	68	0	(0.00 %)
e_4	68	68	0	(0.00 %)
e_5	102	104	2	(1.96 %)
Total	474	478	6	(1.26 %)

Table 2. Detailed result for primary 1.

circuit	# of cells	# of macros	# of regions
primary 1	752	0	6
test 4	1485	4	7

Table 1. Statistical data of benchmark examples

	Vertex weight in G_R $W_R(v_i)$	Vertex weight in G_c $W_c(v_i)$	mismatch cost $\lvert W_R(v_i)-W_c(v_i)\rvert$	
v_1	65594	65540	54	(0.08 %)
v_2	32949	32840	159	(0.48 %)
v_3	33559	33560	1	(0.03 %)
v_4	38441	38340	101	0.26 %)
v_5	42712	42760	48	(0.11 %)
v_6	20990	21000	10	(0.05 %)
v_7	41492	41700	208	(0.50 %)
Total	275737	275740	581	(0.21 %)
	Edge weight in G_R $W_R(e_i)$	Edge weight in G_c $W_c(e_i)$	mismatch cost $\lvert W_R(e_i)-W_c(e_i)\rvert$	
e_1	89	89	0	(0.00 %)
e_2	100	100	0	(0.00 %)
e_3	194	195	1	(0.51 %)
e_4	194	206	12	(0.62 %)
e_5	166	166	0	(0.00 %)
e_6	122	164	42	(34.4 %)
e_7	89	89	0	(0.00 %)
e_8	200	147	53	(26.5 %)
Total	1154	1156	108	(9.3 %)

Table 3. Detailed result for test 4.

	Vertex weight in G_R $W_R(v_i)$	Vertex weight in G_c $W_c(v_i)$	mismatch cost $\lvert W_R(v_i)-W_c(v_i)\rvert$	
v_1	2400	2400	0	(0 %)
v_2	2400	2400	0	(0 %)
v_3	2400	2400	0	(0 %)
v_4	2400	2400	0	(0 %)
v_5	2400	2400	0	(0 %)
v_6	2400	2400	0	(0 %)
Total	14400	14400	0	(0 %)
	Edge weight in G_R $W_R(e_i)$	Edge weight in G_c $W_c(e_i)$	mismatch cost $\lvert W_R(e_i)-W_c(e_i)\rvert$ if $W_c(e_i) > W_R(e_i)$ otherwise 0	
e_1	8	9	1	(12.5 %)
e_2	8	3	0	(0 %)
e_3	8	3	0	(0 %)
e_4	8	9	1	(12.5 %)
e_5	8	8	0	(0 %)
e_6	8	4	0	(0 %)
e_7	8	5	0	(0 %)
e_8	8	7	0	(0 %)
e_9	8	10	2	(25 %)
e_{10}	8	6	0	(0 %)
e_{11}	8	1	0	(0 %)
e_{12}	8	2	0	(0 %)
Total	96	67	4	(4.2 %)

table 4. Detailed result of the placement of a circuit having 144 cells on six faces of 3-D box.