TR-87-015

FAST ALGORITHM FOR OPTIMAL LAYER ASSIGNMENT

TR-87-015

FAST ALGORITHM FOR OPTIMAL LAYER ASSIGNMENT

Y. S. Kuo, T. C. Chern and Wei-Kuan Shih

Institute of Information Science

Academia Sinica

Address for correspondence:

Dr. Yue-Sun Kuo

Institute of Information Science

Academia Sinica

Nankang, Taipei, Taiwan

The Republic of China

Keywords

layer assignment

maximum cut

planar graph

routing

via minimization

Abstract

Given the geometry of wires for interconnections, we want to assign two conducting layers to the segments of these wires so that the number of vias required is minimized. This layer assignment problem, also referred to as the via minimization problem, has been formulated as finding a maximum cut of a planar graph. In this paper, we propose a new algorithm for optimal layer assignment under a general model where the planar graph has real-valued edge weights. The time complexity of the proposed algorithm is $O(n^{3/2} \log n)$ where n is the number of wire-segment clusters in a given layout. In contrast, all existing optimal algorithms for layer assignment have the time complexity of $O(n^3)$. And none of these existing algorithms can find an optimal layer assignment under such a general model.

1. Introduction

We consider the layer assignment problem in an environment where two conducting layers are used for interconnection. Most existing routing algorithms for two layers assign all the vertical wire segments to one layer and all the horizontal wire segments to the other. Therefore, a large number of vias are introduced to interconnect the wire segments on different layers. Vias not only reduce the reliability and performance of the circuit, but also increase the manufacturing cost. Thus, it is desirable to reduce the number of vias. The objective of the layer assignment problem is to assign wire segments to the layers so that the number of vias is minimized. (This problem is also known as the via minimization problem.)

The layer assignment problem can be stated as follows: Given a collection of nets such as in Fig. 1 (borrowed from [18]), each net consisting of wire segments that electrically connect a set of terminals, find a layer assignment to all the wire segments such that any two wire segments in different nets that cross or overlap each other (The design rules must be taken into account.) are assigned to different layers. When two connected wire segments are assigned to different layers, a via must be introduced. Then minimizing the number of vias is equivalent to minimizing the total layer changes in the layout. A possible layer assignment for the nets in Fig. 1 is shown in Fig. 2 where 3 vias are introduced for layer changes.

Hashimoto and Stevens first formulated the layer assignment problem as a graph-theoretic maximum cut problem [9]. By using a similar but more general graph model, Stevens and VanCleemput proposed an approximate method [19]. Ciesielski and Kinnen proposed an integer programming method, but the time complexity of their algorithm is exponential [4]. Kajitani identified the wire-segment clusters in a layout, and showed that the graph in Hashimoto's model is planar [12]. Thus optimal polynomial-time algorithms for via minimization were proposed based on the maximum cut algorithms for planar graphs [3][12][18]. Recently, Chang and Du developed a heuristic algorithm by splitting vertices in a graph [2].

All existing polynomial-time algorithms [3][12][18] for optimal layer assignment are based on Hadlock's maximum cut algorithm for planar graphs [1] [8]. Since Hadlock's algorithm includes subroutines for finding all-pair shortest paths and for finding a maximum weight matching of a dense graph, these layer assignment algorithms are very involved and have the time complexity of $O(n^3)$ where n is the number of wire-segment clusters in the given layout. In contrast, we shall present an $O(n^{3/2} \log n)$ algorithm for optimal layer assignment which is faster and more general. The proposed algorithm is based on a model due to Pinter [18]. Note that the original graph model of Hashimoto and Stevens is rather restricted. By introducing negative weights associated with the edges of the planar graph, Pinter generalized their model to allow wire segments of any orientations (not necessarily horizontal or vertical) and to allow 3-way split points (T-shape connections) in the nets. We shall show that Hadlock's algorithm does not work under this general model (The opposite was claimed in [18].) but our algorithm does.

In Pinter's model, a split point is connected to at most 3 wire segments, i.e. its split number is at most 3. This limitation can be easily removed by adopting the modeling methods in [3][19] though the optimality of the proposed algorithm must be compromised. Since split points with split numbers greater than 3 are uncommon in a layout, it is conceivable that this algorithm can get near-optimal solutions with such extensions.

The layer assignment problem considered in this paper is sometimes referred to as a constrained via minimization problem since the geometry of a layout is given and fixed. The problem in which both topology of the layout and the layer assignment are to be decided is referred to as an unconstrained via minimization problem [2][11][16].

In the next section, we briefly review Pinter's graph model for layer assignment. In Section 3, a series of problem transformations are introduced to show that finding a maximum cut of a planar graph can be reduced to finding a minimum complete matching of a sparse graph. The algorithm for optimal layer assignment is then presented in Section 4 with emphasis on a recursive procedure for finding the minimum complete matching. Finally, in Section 5, we will make some remarks on why Hadlock's algorithm can not find a maximum cut under Pinter's general model.

2. The Graph Model

In this section, we briefly describe Pinter's graph model for layer assignment [18]. The emphasis is placed on the intuition behind the model.

In a given layout such as in Fig. 1, one can identify the following objects:

A via candidate is a maximal piece of wire that does not cross or overlap any other wire, and can accommodate at least one via.

A wire segment is a piece of a wire connecting two via candidates.

A wire-segment cluster (or simply cluster) is a maximal set of mutually crossing or overlapping wire segments.

For example, in Fig. 1, wire segments are labeled by numbers 1 through 18. The wire segments 4,5,6,12 and 14 form one cluster; the wire segments 7,8,9 and 18 form another cluster. Connecting the two clusters are 3 via candidates.

Note that in each cluster, once a wire segment is assigned to a certain layer, layer assignment of the rest of the cluster is forced. Thus there are only two possible ways to assign the wire segments in a cluster to layers. With a prescribed layer assignment, a cluster is said to be flipped over if all the wire segments in the cluster are reassigned to the opposite layers.

The clusters can form a planar graph called the cluster graph. Each vertex of the graph corresponds to a cluster, and two vertices are connected by an edge iff their corresponding clusters are connected to at least a common via candidate. The cluster graph for the layout in Fig. 1 is shown in Fig. 3 where each vertex is labeled by using a representative in its corresponding cluster.

Assume that a layer assignment such as in Fig. 2 is known. Then associated with each edge e of the cluster graph is a weight w(e) defined as follows: Let v be the number of via candidates connecting the two clusters incident to e, and let σ be the number of vias introduced by the known layer assignment connecting the two clusters. Then

$$w(e) = \sigma - (v - \sigma).$$

In other words, the weight indicates the via reduction that can be achieved due to flipping over either one of the two clusters. As an example, in Fig. 3, the weight for the edge connecting clusters 5 and 15 is 2 since for this edge $v = \sigma = 2$ (with reference to Fig. 2). If one of the clusters 5 and 15 is flipped over, then the two vias connecting clusters 5 and 15 can be eliminated.

An arbitrary layer assignment L can be obtained from a known layer assignment L_0 by flipping over a set of clusters. Let $\sigma(L)$ and $\sigma(L_0)$ be the numbers of vias introduced by L and L_0 respectively, and let X be the set of clusters that are flipped over. Then

$$\sigma(L) = \sigma(L_0) - \sum_{e \in E(X,\bar{X})} w(e) \qquad (1)$$

where $E(X,\bar{X})$ is a cut separating X and $\bar{X}$, i.e. the set of edges connecting vertices in X and vertices not in X.* (1) is due to the fact that for any two clusters both in X or both in $\bar{X}$, the via count between the two clusters remains unchanged, but for two clusters one in X and one in $\bar{X}$, the via count is reduced by w(e). In order to minimize the via count $\sigma(L)$, we want to find a cut $E(X,\bar{X})$ which maximizes its weight $\sum_{e \in E(X,\bar{X})} w(e)$, i.e. to find a maximum cut. Note that the edge weights w(e) can be positive or negative, but a maximum cut always has nonnegative weight since X can be ϕ and $\sum_{e \in E(X,\bar{X})} w(e) = 0$ for $X = \phi$. In case that a maximum cut has weight 0, L_0 is an optimal layer assignment.

For the cluster graph in Fig. 3, vertex sets {2,5,8} and {11,15} determine a maximum cut of weight 2. Thus an optimal layer assignment can be obtained from the layer assignment shown in Fig. 2 by flipping over clusters 11 and 15. The resulting layer assignment is shown in Fig. 4.

3. Problem Transformations

Let $G = (V,E)$ be the cluster graph for a given layout. Then G is planar and each edge of E has an associated real-valued weight. By introducing a series of transformations, we will show that a maximum cut of G can be found by finding a minimum complete matching in a certain graph G' constructed from G.

Without loss of generality, we assume that G is connected. (Otherwise a maximum cut can be found by finding maximum cuts in individual connected components.) We first triangulate G by adding some new edges. A triangulation $G_t = (V,E_t)$ of G is a connected planar graph embedded in the plane satisfying

(i) $E \subseteq E_t$,

(ii) Each vertex of G_t has degree at least 2,

(iii) Each face of G_t is enclosed by a simple cycle of three edges, and

(iv) Any two faces of G_t share at most one edge.

We assign zero weight to each new edge in $E_t - E$. As an example, a traingulation of the planar graph in Fig. 3 is shown in Fig. 5.

Lemma 1. A maximum cut of $G = (V,E)$ corresponds to a maximum cut of $G_t = (V,E_t)$, and vice versa.

Consider a geometric dual $G_d = (V_d,E_d)$ of $G_t = (V,E_t)$ [5]. G_d

can be constructed from G_t as follows: Consider an embedding of G_t in the plane. Associated with each face of G_t, there is a vertex in G_d. For each edge shared by two faces of G_t, there is an edge in G_d connecting the two corresponding vertices. A geometric dual of the planar graph shown in Fig. 5 is illustrated in Fig. 6. We assign to each edge of E_d the same weight as its corresponding edge of E_t. In general, a geometric dual of a planar graph is a multigraph. However, due to the construction of G_t, G_d contains no self-loops and parallel edges, and G_d is regular.

Lemma 2. $G_d = (V_d, E_d)$ is a cubic planar graph. (A graph is cubic if each vertex of the graph is of degree 3.)

Proof: G_d contains no self-loops and parallel edges since G_t satisfies (ii) and (iv). Thus G_d is a graph. Each vertex of V_d is of degree 3 since G_t satisfies (iii). The planarity of G_d is due to the fact that G_d is a geometric dual of G_t. Q.E.D.

In $G_d = (V_d, E_d)$, an edge set $D \subseteq E_d$ is said to be even-degree if each vertex of V_d is incident to an even number of edges in D. The weight of an even-degree edge set D is the total weight of the edges in D. Since $G_d = (V_d, E_d)$ is a geometric dual of $G_t = (V, E_t)$, there is a one-to-one correspondence between edges of E_t and edges of E_d. This induces a natual correspondence between the cuts of G_t and the even-degree edge sets of G_d.

Theorem 1. A cut of $G_t = (V, E_t)$ corresponds to an even-degree edge set of $G_d = (V_d, E_d)$, and vice versa.

Theorem 1 will be proved later in the appendix. Here we simply illustrate this theorem by an example. For the graph in Fig. 5, consider the vertex set $X = \{8,11\}$. X determines a cut $E(X,\bar{X})$ where

$$E(X,\bar{X}) = \{(8,2),(8,5),(8,15),(11,2),(11,5),(11,15)\}.$$

In the dual graph (Fig. 6), the edge set that corresponds to $E(X,\bar{X})$ is

$$\{(E,D),(D,A),(A,E),(F,C),(C,B),(B,F)\}.$$

Apparently, this edge set is even-degree, and consists of two simple cycles.

Corollary 1. A maximum cut of $G_t = (V,E_t)$ corresponds to a maximum (weight) even-degree edge set of $G_d = (V_d, E_d)$, and vice versa.

Proof: Directly from Theorem 1.

The following lemma characterizes a maximum even-degree edge set of G_d.

Lemma 3. Let D be a maximum even-degree edge set of $G_d = (V_d, E_d)$. Then D is either empty or a union of vertex-disjoint nonnegative cycles.

Proof: Assume $D \neq \phi$. Since G_d is a cubic graph, each vertex of G_d is adjacent to 0 or 2 edges in D. Thus D is a union of vertex-disjoint cycles in G_d. The claim then follows from the fact that D is maximum.

Q.E.D.

To find a maximum even-degree edge set of $G_d = (V_d, E_d)$, we construct a graph $G' = (V', E')$ from G_d. Each vertex v of G_d is replaced by a "star" in G' and each edge e of G_d has a surrogate in G' as depicted in Fig. 7. For the cubic planar graph in Fig. 6, the constructed graph is illustrated in Fig. 8. Define the edge weights of G' as follows: the surrogate of each edge $e \in E_d$ has the same weight as e; and all new edges in stars have zero weights. Similar constructions have appeared in [13][17].

A matching M of graph $G = (V', E')$ is a set of edges no two of which have a common vertex. If $|M| = |E'|/2$, then M is called a complete matching. A maximum weight matching (minimum complete matching) is a matching (complete matching) of G' whose total weight is maximum (minimum).

Theorem 2. Let $M \subseteq E'$ be a minimum complete matching of $G' = (V', E')$. Then $E_d - M$ is a maximum even-degree edge set of $G_d = (V_d, E_d)$.

Proof: Let $M \subseteq E'$ be any complete matching of G'. If M contains edge (v',v") in a star substituting a vertex v of G_d (see Fig. 7), then M must contain all the edges incident to v in G_d and hence v has degree 0 in the subgraph of G_d induced by $E_d - M$. On the other hand, if M does not contain (v',v"), then v has degree 2 in the subgraph. Thus $E_d - M$ is an even-degree edge set of G_d. Conversely, let D be any even-degree edge set of G_d. As shown in Lemma 3, D is either empty or a union of vertex-disjoint cycles in G_d. Thus from the construction of G', one can observe that there exists a complete matching M of G' such that $D = E_d - M$. (Such a complete matching is illustrated in Fig. 8 with respect to an even-degree edge set shown in Fig. 6.) Clearly the weight of $E_d - M$ is maximum if and only if the weight of M is minimum. Q.E.D.

4. The Algorithm

Let us summarize the algorithm for optimal layer assignment and analyze its time complexity.

Input: The cluster graph $G_c = (V_c, E_c)$ for a given layout. Real-valued weights are assigned to the edges of G_c with respect to a known layer assignment.

Output: A maximum cut of G_c.

Algorithm MaxCut

1. Decompose G_c into connected components. For each connected component $G = (V,E)$, do steps 2 to 5.
2. Construct a triangulation $G_t = (V, E_t)$ of $G = (V,E)$ by adding new edges to G.
3. Construct a geometric dual $G_d = (V_d, E_d)$ of $G_t = (V, E_t)$.
4. Construct the graph $G' = (V', E')$ from G_d. Each vertex of G_d is replaced by a "star" in G'.
5. Find a minimum complete matching M of G'. M determines a maximum even-degree edge set $E_d - M$ of G_d which corresponds to a maximum cut of G_t and thus a maximum cut of $G = (V,E)$.
6. Combining the maximum cuts for individual connected components, we have a maximum cut for $G_c = (V_c, E_c)$.

Let n_c and n be the numbers of vertices of G_c and G respectively. Since G_c is planar, due to Euler's Theorem [5], G_c has $O(n_c)$ edges. Thus Step 1 can be computed in $O(n_c)$ time by using a simple depth-first search. Applying Euler's Theorem repeatedly, we can see that each of the graphs G, G_t and G_d has O(n) vertices and O(n) edges. G' also has O(n) vertices and O(n) edges for each star in G' contains 5 vertices and 7 edges. Each of Step 2 and Step 3 takes O(n) time since they can be carried out by embedding G and G_t in the plane and identifying their faces [10]. That Step 4 takes O(n) time and Step 6 takes $O(n_c)$ time is obvious. We have argued that all steps except Step 5 takes linear time in the worst case. Below we shall describe an $O(n^{3/2} \log n)$ algorithm for Step 5 of Algorithm MaxCut.

A minimum complete matching of G' = (V',E') can be found by finding a maximum weight matching of the same graph except that the weight w(e) of each edge $e \in E'$ must be replaced by a new weight W - w(e) where W is a large constant. (The negation of w(e) converts a minimization problem to a maximization problem. The added large constant W forces the obtained matching to be complete.) Lipton and Tarjan have presented an $O(n^{3/2} \log n)$ algorithm for finding a maximum weight matching of a planar graph by applying the planar separator theorem [14][15]. For graph G' = (V',E') which is not always planar, the same "divide-and-conquer" method can still be applied as K. Matsumoto, et al. have pointed out [17].

Input: Graph G" = (V",E"). G" is the same as G' = (V',E') except that
the weight w(e) of each edge $e \in E'$ is replaced by W - w(e)
where W is a large constant.

Output: A maximum weight matching of G".

Algorithm MaxMatching

1. If G" contains a few (no more than a fixed constant) vertices, find a maximum weight matching of G" by the algorithm in [6].
2. Otherwise, partition the vertices of G" into three sets A, B, and C such that no edge joins a vertex in A with a vertex in B, $|A|$, $|B| \le c_1|V"|$ and $|C| \le c_2|V"|^{1/2}$ for suitable constants c_1 (< 1) and c_2.
3. Let G_A and G_B be the subgraphs of G" induced by A and B respectively. Apply the algorithm recursively to find maximum weight matchings M_A in G_A and M_B in G_B. Let $M = M_A \cup M_B$ and $S = A \cup B$.
4. Add C one vertex at a time to S. Each time a vertex is added to S, replace M by a maximum weight matching in G_s, the subgraph of G" induced by S. Stop when S = V".

Lemma 4. A minimum complete matching of G' = (V',E') can be found in $O(n^{3/2} \log n)$ time by using Algorithm MaxMatching.

Proof: In Algorithm MaxMatching, Step 1 takes constant time. The partitioning in Step 2 is guaranteed by the planar separator theorem and takes O(n) time [14]. Step 3 involves two recursive calls on subgraphs of size at most $c_1 n$. Step 4 takes $O(n^{3/2} \log n)$ time since each updating of M takes $O(n \log n)$ time and $|C| = O(n^{1/2})$ [6]. Solving the recurrence relation

$$T(n) = T(n_1) + T(n_2) + O(n^{3/2} \log n)$$

where $n_1 + n_2 \le n$ and $n_1, n_2 \le c_1 n$, we have $T(n) = O(n^{3/2} \log n)$. (For a detailed analysis of Algorithm MaxMatching, please see [17].)

Q.E.D.

Theorem 3. Given a cluster graph $G_c = (V_c, E_c)$ with real-valued edge weights, Algorithm MaxCut can find a maximum cut of G_c in $O(n_c^{3/2} \log n_c)$ time. In other words, an optimal layer assignment can be found in $O(n_c^{3/2} \log n_c)$ time where n_c is the number of clusters in a given layout.

5. Concluding Remarks

We have presented a new algorithm for optimal layer assignment. The algorithm has the time complexity of $O(n^{3/2} \log n)$ where n is the number of clusters in a given layout. In contrast, all existing polynomial-time algorithms for optimal layer assignment are based on Hadlock's maximum cut algorithm for planar graphs which has the time complexity of $O(n^3)$ [8].

The new algorithm is not only more efficient, but also more general. It can find an optimal layer assignment under Pinter's general model ([18]) while Hadlock's algorithm can not. In Pinter's model for layer assignment, the cluster graph can have negative weights associated with its edges. In the following, we will see why Hadlock's algorithm can not find a maximum cut in a planar graph with negative weights. Hadlock tried to find a maximum cut of a planar graph by finding a maximum even-degree edge set in the graph's geometric dual (Similar to Corollary 1). His algorithm relies on the result that the complement of a maximum even-degree edge set is a union of edge-disjoint shortest paths connecting pairs of odd-degree vertices. This result is true for planar graphs with nonnegative weights, however, is not true for planar graphs with negative weights. As an example, for the planar graph in Fig. 3, its geometric dual is shown in Fig. 9 where the maximum even-degree edge set is a simple cycle. (Edges in the cycle are marked with short bars.) The complement of the maximum even-degree edge set consists of an edge and C plus a negative cycle joining B and C. Note that in the presence of negative cycles, the shortest path is not well-defined, and finding a simple path which is shortest is NP-hard [7]. Therefore, Hadlock's algorithm can not be adapted to the general situation where the planar graph has negative weights.

Appendix

We will prove Theorem 1 in this appendix. Given a connected graph $G = (V,E)$, let $E(A,B)$ denote the set of edges of G that connect two disjoint vertex sets A and B. Thus an edge set $C \subseteq E$ is a cut iff there is a partition of V into two nonempty sets X and $\bar{X}$ such that $C = E(X,\bar{X})$. (We exclude the empty set as a cut for ease of presentation.) A cut is minimal if none of its proper subsets is a cut.

Lemma 5. The union of two disjoint cuts is a cut.

Proof: Let $C = C_1 \cup C_2$ where $C_1 = E(X,\bar{X})$, $C_2 = E(Y,\bar{Y})$ and $C_1 \cap C_2 = \phi$. Consider the vertex sets $X \cap Y$, $X \cap \bar{Y}$, $\bar{X} \cap Y$ and $\bar{X} \cap \bar{Y}$ as shown in Fig. 10. Since $C_1 \cap C_2 = \phi$, we have

$$E(X\cap Y,\bar{X}\cap\bar{Y}) = E(X\cap\bar{Y},\bar{X}\cap Y) = \phi,$$

$$C_1 = E(X\cap Y,\bar{X}\cap Y) \cup E(X\cap\bar{Y},\bar{X}\cap\bar{Y}) \text{ and}$$

$$C_2 = E(X\cap Y,X\cap\bar{Y}) \cup E(\bar{X}\cap Y,\bar{X}\cap\bar{Y}).$$

Thus $C = C_1 \cup C_2$ is a cut separating $(X\cap Y)\cup(\bar{X}\cap\bar{Y})$ and $(X\cap\bar{Y})\cup(\bar{X}\cap Y)$. Q.E.D.

Lemma 6. Let C_1 and C_2 be cuts of G and $C_1 \subseteq C_2$. Then $C = C_2 - C_1$ is also a cut of G.

Proof: let $C_2 = E(X,\bar{X})$ and $C_1 = E(Y,\bar{Y})$. Consider the vertex sets $X\cap Y$, $X\cap\bar{Y}$, $\bar{X}\cap Y$ and $\bar{X}\cap\bar{Y}$ as shown in Fig. 11. Then

$$C_2 = E(X\cap Y,\bar{X}\cap Y) \cup E(X\cap Y,\bar{X}\cap\bar{Y}) \cup E(X\cap\bar{Y},\bar{X}\cap Y) \cup E(X\cap\bar{Y},\bar{X}\cap\bar{Y}).$$

Since $C_1 \subseteq C_2$ and C_1 is a cut separating Y and $\bar{Y}$, we have

$$C_1 = E(X\cap Y,\bar{X}\cap\bar{Y}) \cup E(X\cap\bar{Y},\bar{X}\cap Y) \text{ and}$$

$$E(X\cap Y, X\cap\bar{Y}) = E(\bar{X}\cap Y, \bar{X}\cap\bar{Y}) = \phi.$$

Consequently, $C = C_2 - C_1 = E(X\cap Y, \bar{X}\cap Y) \cup E(X\cap\bar{Y}, \bar{X}\cap\bar{Y})$ and C is a cut separating $(X\cap Y) \cup (\bar{X}\cap\bar{Y})$ and $(X\cap\bar{Y}) \cup (\bar{X}\cap Y)$. Q.E.D.

Lemma 7. An edge set $C \subseteq E$ is a cut of $G = (V,E)$ iff it is a union of disjoint minimal cuts of G.

Proof: Directly from Lemmas 5 and 6.

The even-degree edge sets have essentially the same properties as the cuts. We just state the result without proof.

Lemma 8. An edge set is even-degree iff it is a union of edge-disjoint simple cycles.

In the planar graph theory, the minimal cuts of a planar graph are associated with the simple cycles of its dual graph [5]. The following lemma is a classic result.

Lemma 9. Let $G = (V,E)$ be a connected planar graph, and let $G_d = (V_d, E_d)$ be its geometric dual. Then there is a one-to-one correspondence $f:E \to E_d$ which maps a minimal cut of G to a simple cycle of G_d and vice versa.

Due to Lemmas 7, 8 and 9, Theorem 1 is proven.

References

[1] K. Aoshima and M. Iri, Comments on F. Hadlock's paper: finding a maximum cut of a planar graph in polynomial time, SIAM J. Comput., 6 (1977), pp. 86-87.

[2] K. C. Chang and D. H. Du, Efficient algorithms for layer assignment problem, IEEE Trans. on Computer-Aided Design, vol. CAD-6, no. 1, January 1987, pp. 67-78.

[3] R. W. Chen, Y. Kajitani, and S. P. Chan, A graph-theoretic via minimization algorithm for two-layer printed circuit boards, IEEE Trans. Circuits Syst., vol. CAS-30, no. 5, pp. 284-299, May 1983.

[4] M. J. Ciesielski and E. Kinnen, An optimum layer assignment for routing in IC's and PCB's, Proc. 18th Design Automation Conf., June 1981, pp. 733-737.

[5] S. Even, Graph Algorithms, Computer Science Press, Maryland, 1979.

[6] Z. Galil, S. Micali and H. Gabow, Priority queues with variable priority and an O(EV log V) algorithm for finding a maximal weighted matching in general graphs, 23rd Annual Symposium on Foundations of Computer Science, Chicago, 1982, pp. 255-261.

[7] M. R. Garey and D. S. Johnson, Computers and Intractability, a Guide to the Theory of NP-completeness, W. H. Freeman and Company, San Francisco, 1979.

[8] F. Hadlock, Finding a maximum cut of a planar graph in polynomial time, SIAM J. Computing, vol. 4, no.3, pp. 221-225, Sept. 1975.

[9] A. Hashimoto and J. Stevens, Wire routing by optimizing

channel assignment within large apertures, Proc. 8th Design Automation Workshop, June 1971, pp. 155-169.

[10] J. Hopcroft and R. Tarjan, Efficient planarity testing, JACM, Vol. 21, No. 4 (1974), pp. 549-568.

[11] C. P. Hsu, Minimum-via topological routing, IEEE Trans. Computer-Aided Design, vol. CAD-2, no. 4, pp. 235-246, Oct. 1983.

[12] Y. Kajitani, On via hole minimization of routing in a 2-layer board, Proc. IEEE 1980 Int. Conf. Circuits Computers, June 1980, pp. 295-298.

[13] E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.

[14] R. J. Lipton and R. E. Tarjan, A separator theorem for planar graphs, SIAM J. Appl. Math., 36 (1979), pp. 177-189.

[15] R. J. Lipton and R. E. Tarjan, Applications of a planar separator theorem, SIAM J. Comput, 9 (1980), pp. 615-627.

[16] M. Marek-Sadowska, An unconstrained topological via minimization problem for two-layer routing, IEEE Trans. Computer-Aided Design, vol. CAD-3, no. 3, pp. 184-190, July 1984.

[17] K. Matsumoto, T. Nishizeki and N. Saito, Planar multicommodity flows, maximum matchings and negative cycles, SIAM J. Comput., 15 (1986), pp. 495-510.

[18] R. Y. Pinter, Optimal layer assignment for interconnect, J. VLSI and Computer Systems, vol. 1, No. 2, 1984, pp. 123-137.

[19] K. R. Stevens and W. M. VanCleemput, Global via elimination in generalized routing environment, Proc. Int. Symp. on Circuits and Systems, 1979, pp. 689-692.

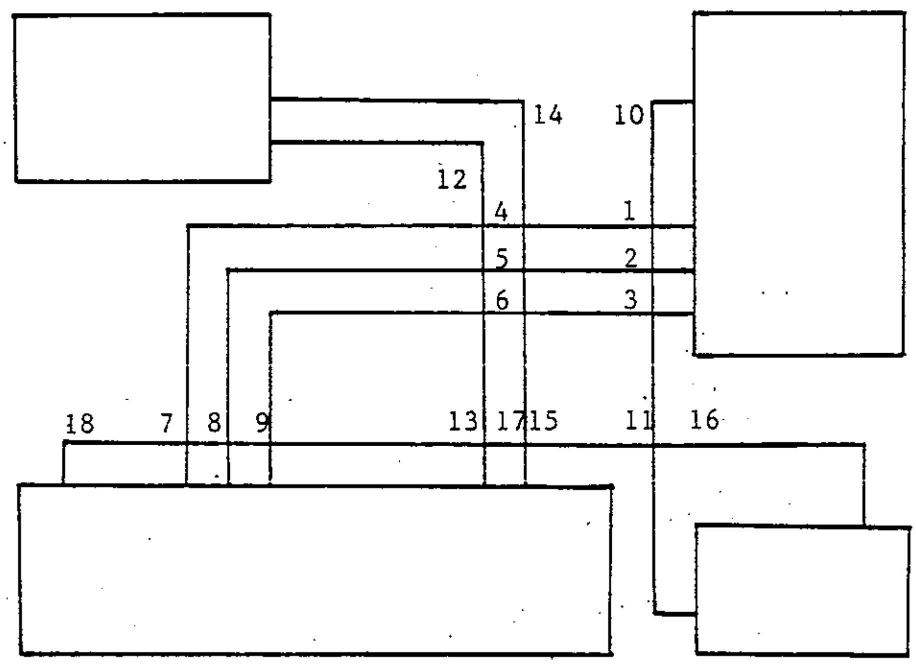


Fig. 1. Layout before layer assignment

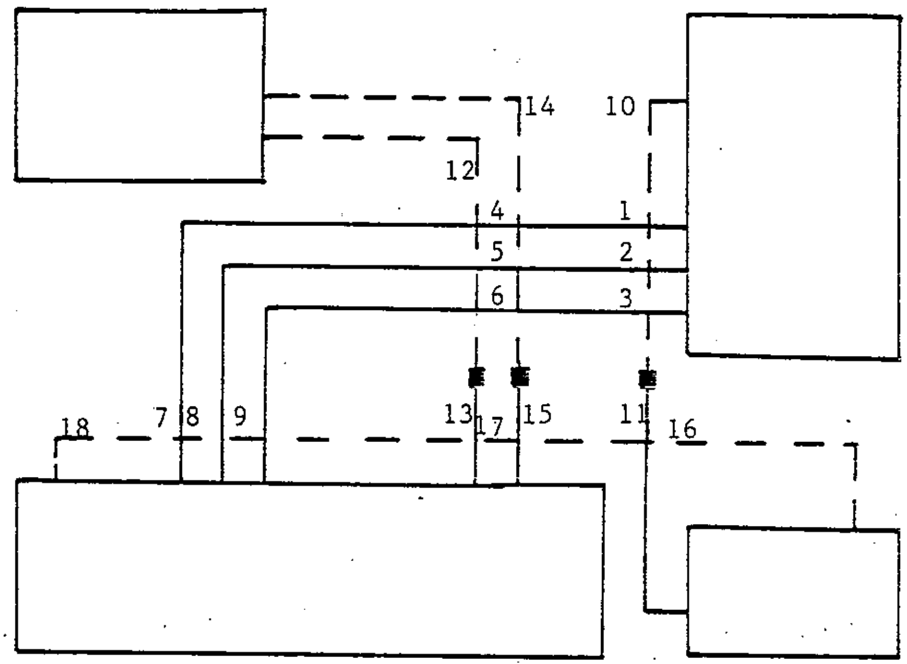


Fig. 2. A possible layer assignment

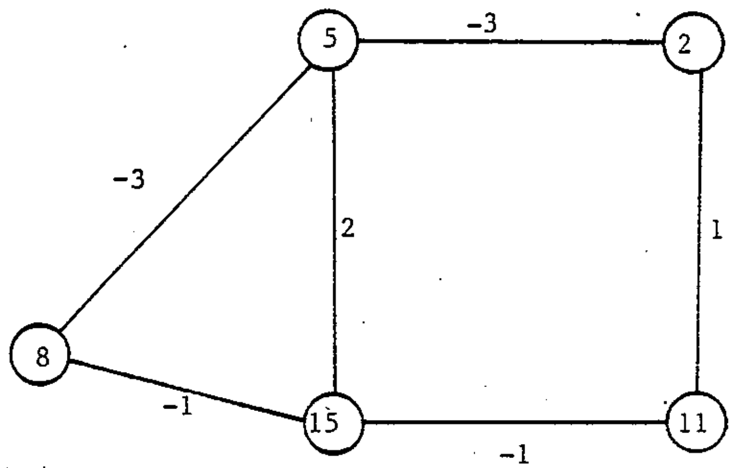


Fig. 3. Cluster graph G

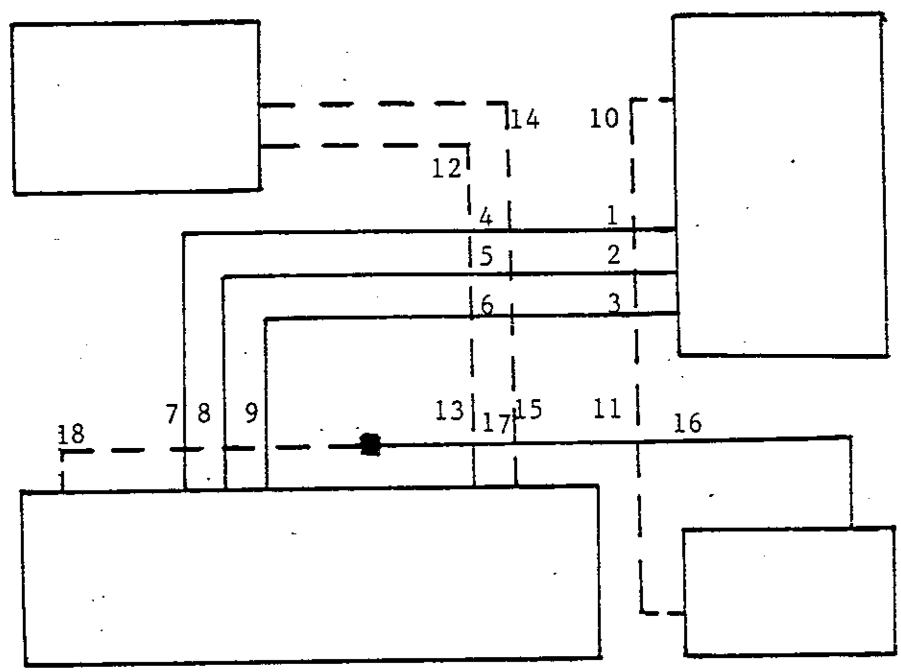
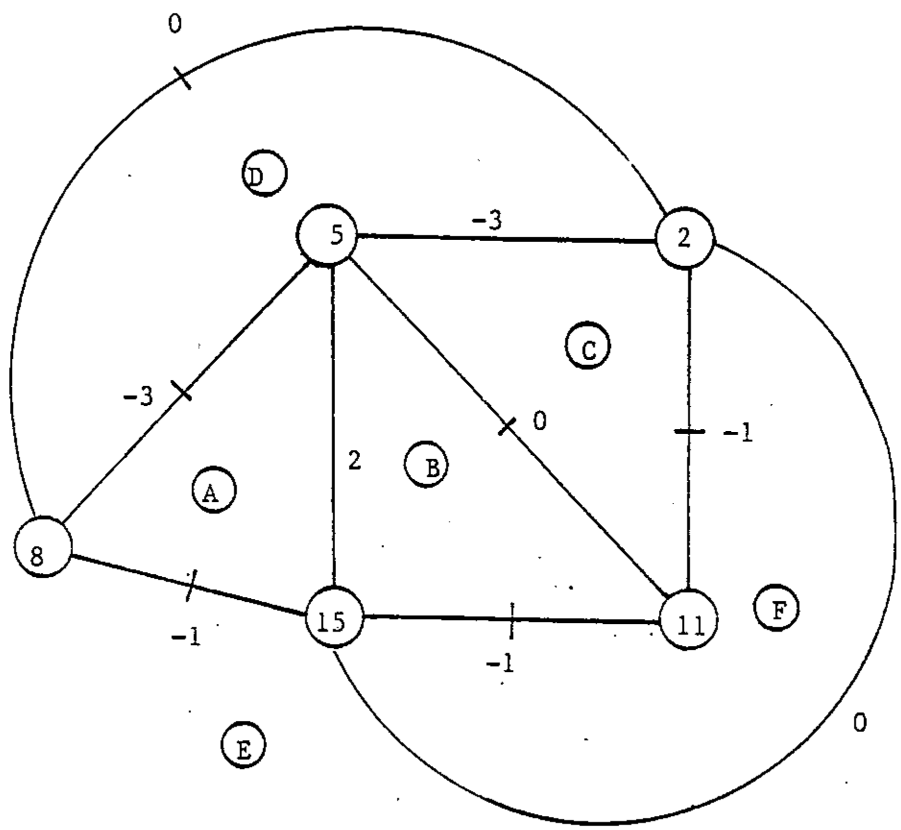
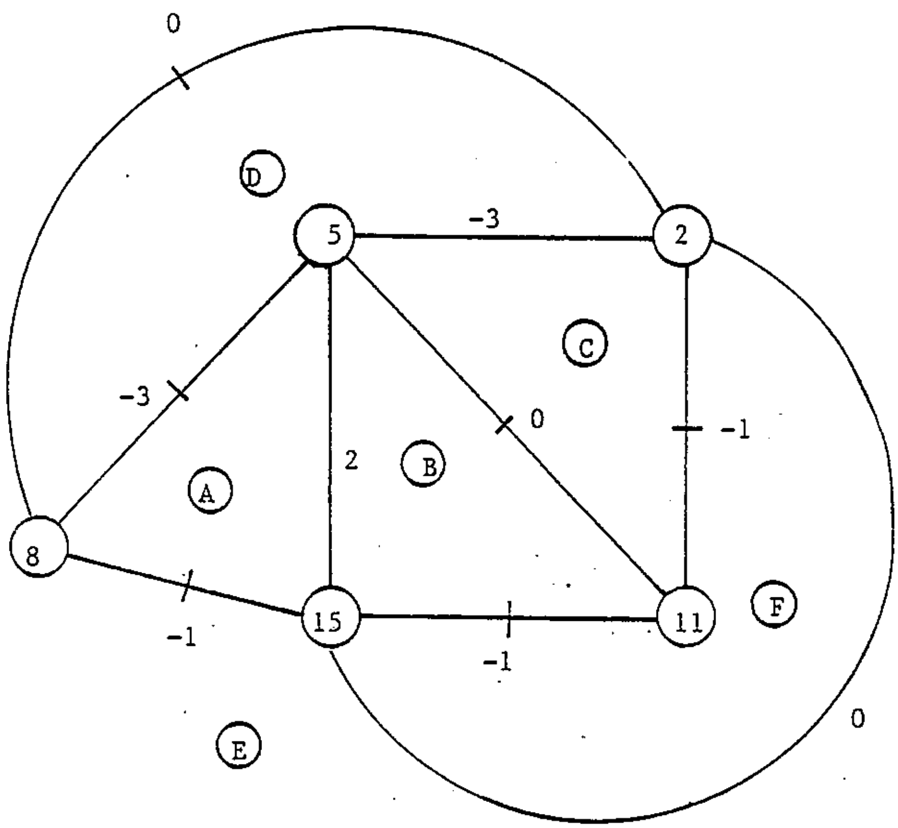


Fig. 4. Optimal layer assignment



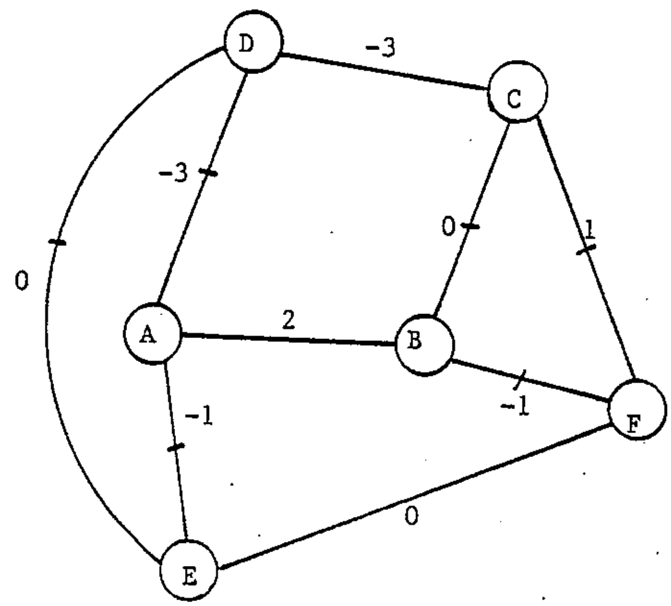
(Edges marked with short bars form a cut.)

Fig. 5. Triangulation G_t of G



(Edges marked with short bars form a cut.)

Fig. 5. Triangulation G_t of G



(Edges marked with short bars
form an even-degree edge set.)

Fig. 6. Geometric dual G_d of G_t

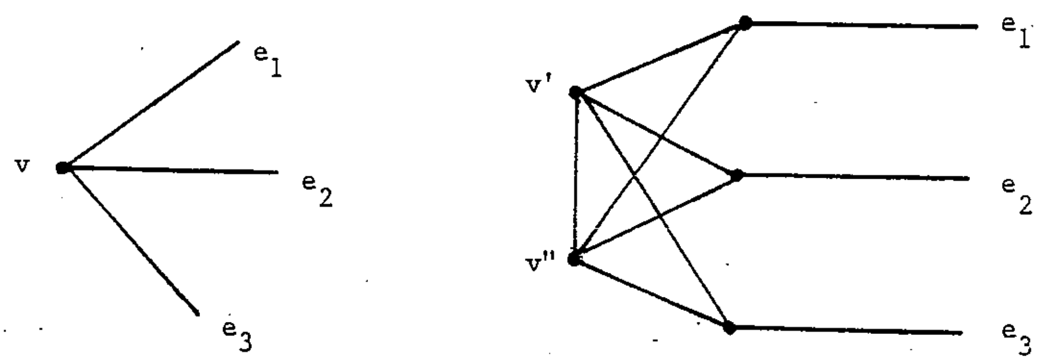
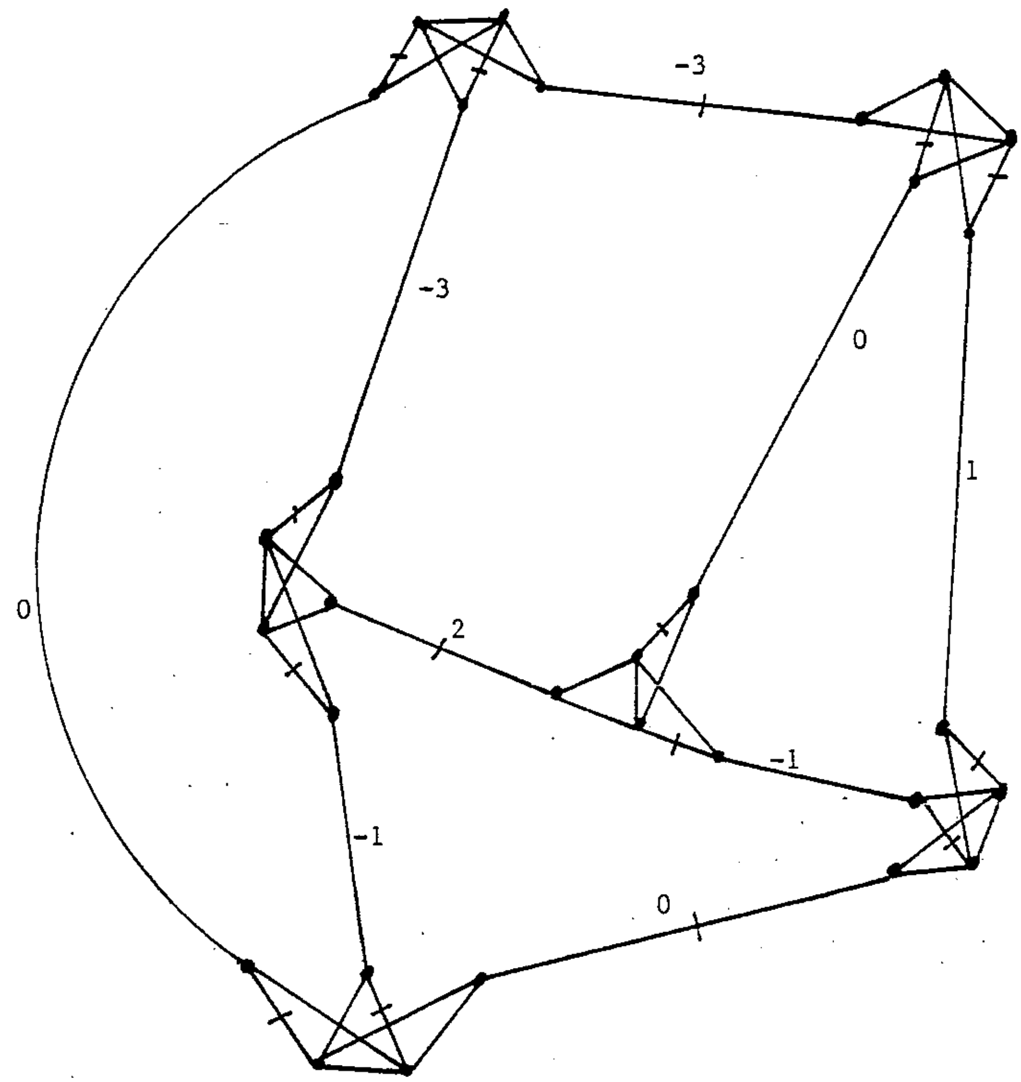


Fig. 7. The star substituting a vertex



(Edges marked with short bars form a complete matching.)

Fig. 8. Graph G' constructed from G_d

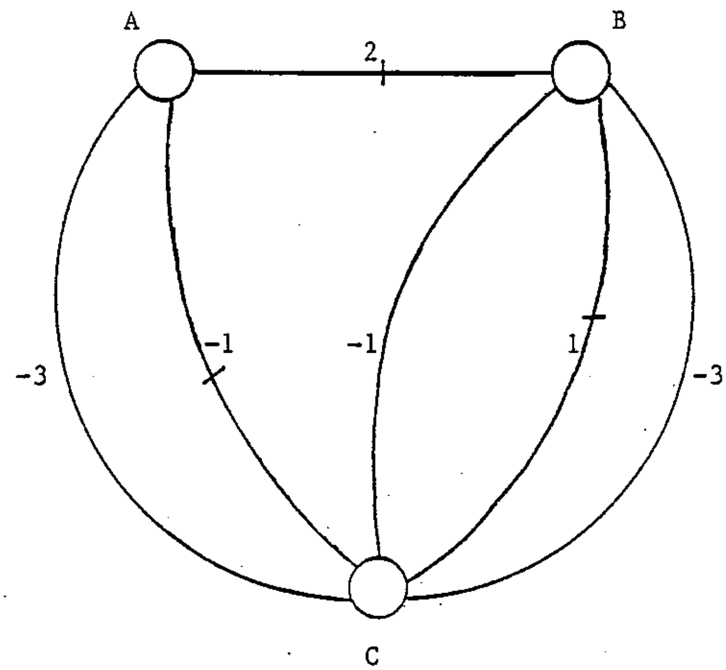


Fig. 9. Geometric dual of G

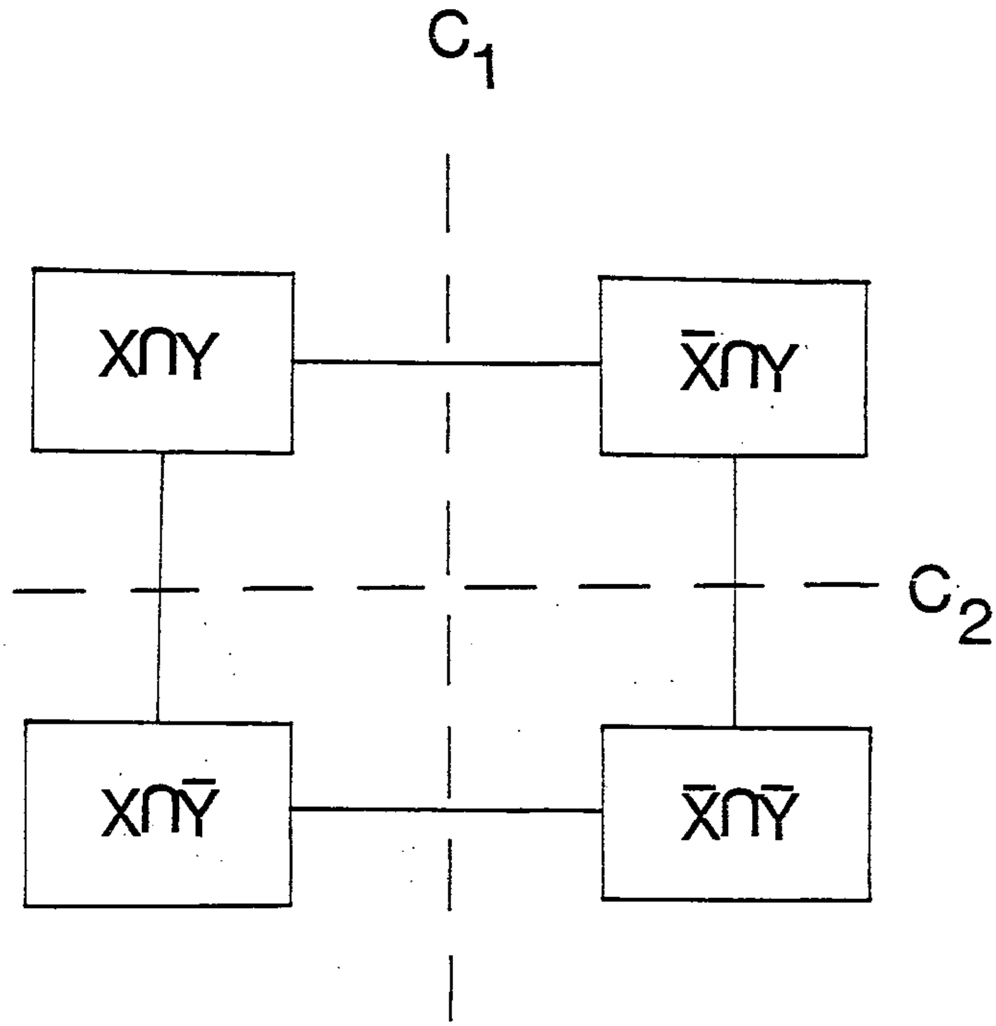


Fig. 10. Illustration for Lemma 5

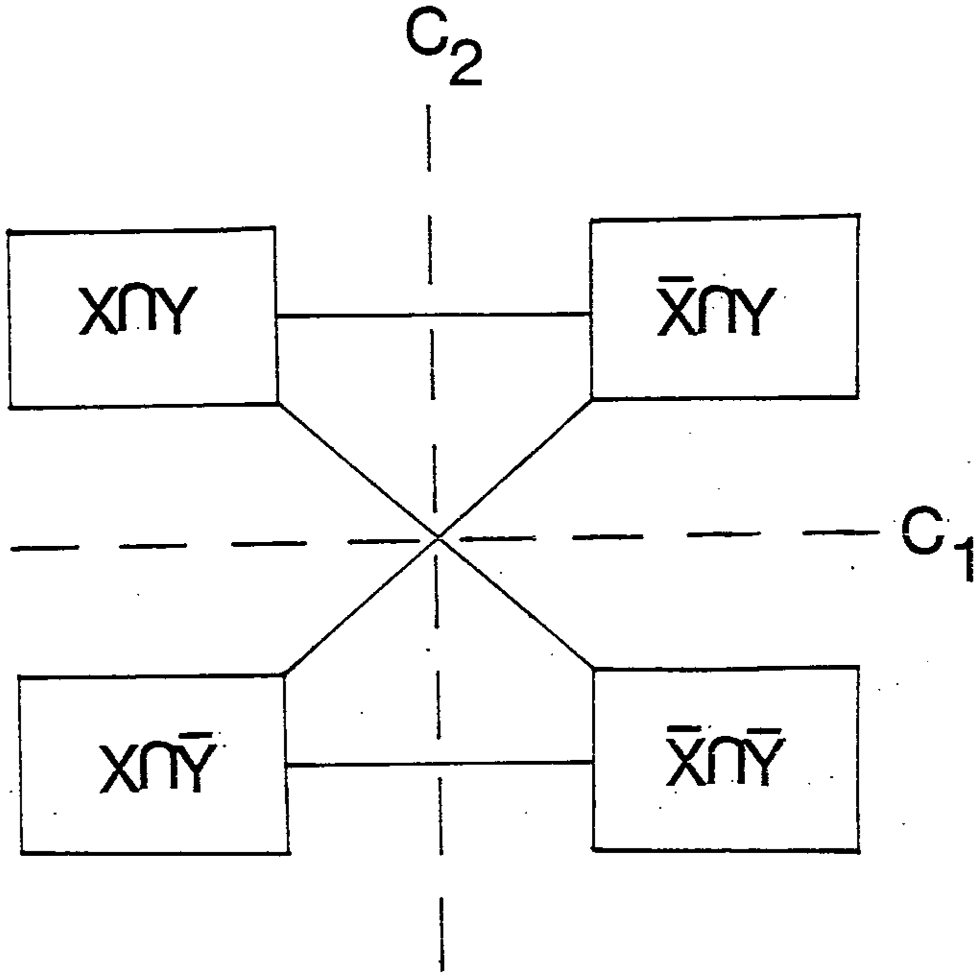


Fig. 11. Illustration for Lemma 6