An Efficient Approach to Constrained Via Minimization for Two-Layer VLSI Routing

Maolin Tang, Kamran Eshraghian and Hon Nin Cheung

Centre for Very High Speed Microelectronic Systems
Edith Cowan University
Perth, WA 6027, Australia

Abstract— Constrained Via Minimization is the problem of reassigning wire segments of a VLSI routing so that the number of vias is minimized. In this paper, a new approach is proposed for two-layer VLSI routing. This approach is able to handle any types of routing, and allows arbitrary number of wire segments split at a via candidate.

1. Introduction

Via is a mechanism (hole) for connecting wire segments of a net distributed on different layers in two-layer or multi-layer VLSI routing. However, in integrated circuit fabrication, the yield is inversely related to the number of vias because a chip with more vias has a smaller probability of being fabricated correctly. In addition, via has an associated resistance that affects circuit performance. Finally, the size of a via is usually larger than the width of wires. As a result, more vias lead to more routing space. Therefore, it is desirable to minimize the number of vias introduced in VLSI routing. However, VLSI routing is a complex and intractable problem. Therefore, existing routers and design tools can only consider the minimization of the number of tracks in channel routing, completion of switchbox routing, and wire length minimization as their primary objectives. As a consequence, via minimization is either ignored or de-emphasized in the routers and design tools, and therefore comes as an 'afterthought' problem in VLSI routing.

Given an initial routing consisting of a set of wire segments, the problem to reassign wire segments to available layers so that the logical connections are maintained and the number of vias required is minimized without changing the topology of the initial routing is so-called *Constrained Via Minimization*, or CVM. Because the via minimization is realized by reassigning the layers of wire segments, it is also referred to as *Layer Assignment Problem*. In this paper, we use the two terms alternatively.

The constrained via minimization is NP-Complete[11]. However, for a special case where the number of the wire segments split at a via candidate is not more than three elegant theoretical results have been obtained [5, 6, 8, 10]. But, the situation that four or more wire segments split at a via candidate can not be avoided in practical VLSI routing. As a result, those methods are not suitable to handle practical routing problems. Although some methods can be used to handle the situation that there are more than three wire segments split at a via candidate, they are only suitable for some particular routing patterns, such as Manhattan routing, and difficult to be adjusted to meet the practical requirements, such as, the layer assignment of special nets [9]. In this paper, a new approach to the two-layer constrained via minimization problem for two-layer VLSI routing is proposed. In contrast to the other existing approaches, this approach is suitable for both grid-based and gridless routing, and arbitrary number of wire segments are permitted to be split at a via candidate. In order to test the efficiency of the heuristic algorithm, we have implemented the heuristic algorithm in C++ on SunSPACR 20. Experimental results showed that this approach is effective and efficient.

This paper is organized as follows: firstly, we present a new graphic representation of a two-layer routing, termed LAP graph. Then, on the basis of that representation, we formulate the constrained via minimization problem as a so-called *Switching Problem*. Next, a heuristic algorithm is proposed for solving the switching problem followed by the issues of practical considerations, and then the experimental results are provided. Finally, we conclude this new approach.

2. LAP Graph

In order to facilitate the description of the LAP graph, the following definitions are introduced.

Planar representation: The projection of a two-layer routing on a planar which parallels to the two routing layers.

Net: A collection of wire segments that electrically connect a set of terminals.

Cross point: A point at which two different wire segments intersect or overlap on the planar representation.

Via: A mechanism (hole) for connecting wire segments of a net on different routing layers.
Via candidate: A proper point at which a via might be introduced. The following two rules are used to select via candidates on the planar representation of a routing. Firstly, a cross point cannot be chosen to be a via candidate; secondly, if there are several contiguous points on a net that are not cross points, we only choose one of them as a via candidate rather than all of them. In fact, we select the one with maximal split degree as a via candidate.
Net segment: A piece of net separated by via candidates. Note that a net segment is not limited to a straight line segment. It can be a collection of connected line segments.
Cluster: A maximal set of mutually crossing or overlapping net segments. Note that once a net segment is assigned to a layer, the layer assignments of the rest net segments in the cluster are determined. Because the net segment can be assigned onto either of the two available routing layers, there are only two possible ways to assign the net segments in a clusters to layers. The layer assignment of a net segment in a cluster only restricts the layer assignments on the net segments in the same cluster and never restricts the layer assignments of the net segments in the other clusters.

Figure 1(a) shows an initial two-layer routing, where dotted lines represent the wire segments on one layer, solid lines represent the wire segments on the other layer, and dots represent vias introduced in this routing; Figure 1(b) shows the corresponding planar representation. On the figure, $v_1, v_2, \ldots, v_7$ are via candidates which are marked as circles, and $n_1, n_2, \ldots, n_{14}$ are net segments; Figure 1(c) and 1(d) are two feasible layer assignments of the net segments. The entire clusters are as follows:

$c_1 = \{n_1\}$; $c_2 = \{n_2, n_8\}$; $c_3 = \{n_3, n_{10}, n_{12}\}$;
$c_4 = \{n_4, n_6, n_7, n_9, n_{14}\}$; $c_5 = \{n_5\}$;
$c_6 = \{n_{11}\}$; $c_7 = \{n_{13}\}$.

Assume that $c_1, c_2, \ldots, c_n$ are entire clusters, $v_1, v_2, \ldots, v_p$ are all via candidates, and R is a feasible layer assignment with regard to a constrained via minimization problem. The LAP graph, G(V, E), is constructed in the following way:

$V = CL \cup VC$, where CL=$\{c_1, c_2, \ldots, c_n\}$, VC=$\{v_1, v_2, \ldots, v_p\}$; $< u_1, u_2 > \in E$ if and only if either of the following conditions is satisfied:

1. $u_1 \in VC$, $u_2 \in CL$, and via candidate u_1 is adjacent to a net segment in the cluster u_2 and the net segment is assigned to the solid line layer in R;
2. $u_1 \in CL$, $u_2 \in VC$, and via candidate u_2 is adjacent to a net segment in the cluster u_1 and the net segment is assigned to the dotted line layer in R.

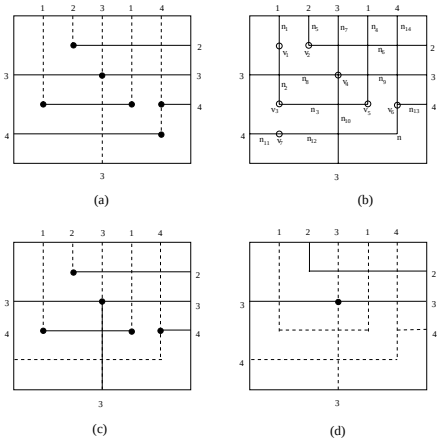

FIGURE 1. Initial routing, planar representation, feasible layer assignment and optimized layer assignment

Figure 2 is the LAP graph corresponding to the feasible layer assignment in Figure 1(c).

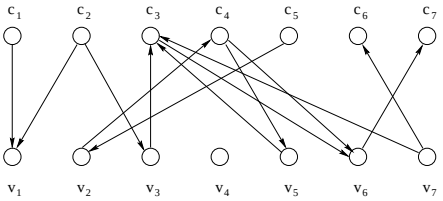

FIGURE 2. LAP graph

LAP graph is a bigraph. It reflects the mutual constraints among the via candidates and clusters. It can be seen from the above two figures that if all net segments being incident to a via candidate are arranged on the same layer, then no via is introduced at the via candidate. In contrast, if there exist two of the wire segments are arranged on two different layers, a via must be introduced at the via candidate in order to connect the net segments on different layers. Reflected on the corresponding LAP graph, the above situation is that if the directions of all the arcs being incident to the via candidate vertex are identical, e.g. all the arcs point to the via candidate, then no via is introduced at the corresponding via candidate; otherwise, a via must be introduced there. Thus, the number of vias introduced in a layer assignment equals to the number of the via candidate vertice on the corresponding LAP graph whose both in-degree and out-degree are not zero. If the degree of a via candidate is zero, that means all the net segments being incident to the via candidate belong to the same cluster. We denote this kind of via candidates as *intrinsic via candidates*. If there is a via introduced in the original routing, then there will be a via in the optimized layer assignment; if there is no via introduced there, then there will be no via introduced in the optimized layer assignment.

3. Problem Formulation

We define *Switching Graphs* of the LAP graph as the graphs obtained by reversing the direction of all the arcs being incident to the vertice in a subset of CL, S. It has been seen that a switching graph corresponds to a feasible layer assignment. The layer assignment is the one that is obtained by switching all the layer assignments of the net segments in S. In fact, the layer assignments and the switching graphs are one-to-one correspondence. Since there are 2^n different switching graphs of the LAP graph, there are 2^n different layer assignments. Thus, the constrained via minimization problem is transferred into the following problem:

Given a LAP graph, find a switching graph of it such that the number of the via candidate vertice whose neither in-degree nor out-degree is zero is minimum. This is named as switching graph problem.

Considering the high computational complexity of the switching problem, a heuristic algorithm is proposed to the intractable problem.

4. Heuristic Algorithm for the Switching Problem

The associating situations between cluster vertex c_i and via candidate vertex v_j are divided into the nine types shown in Figure 3, excluding isomorphic situations. The isomorphic situations refer to the patterns obtained by reversing the direction of all the arcs in the figure.

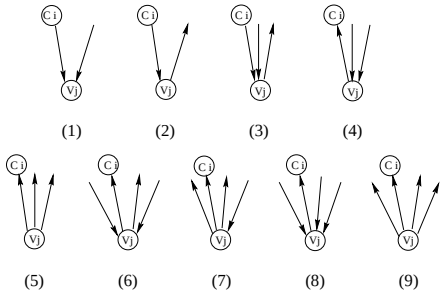

FIGURE 3. The associating situations between cluster vertex c_i and via candidate vertex v_j

A weight is assigned to each of the nine associating types. The weight $w(c_i, v_j)$ is a measurement of the contribution to eliminate the via at the via candidate v_j if we switch the cluster c_i. It is defined as follows:

$$w(c_i, v_j) = \begin{cases} -1 & \text{types (1), (5) and (9)} \\ 0 & \text{types (3), (6) and (7)} \\ 1 & \text{types (2), (4) and (8)} \end{cases} \quad (1)$$

where $< c_i, v_j > \in E$ or $< v_j, c_i > \in E$.

We further define the weight of a cluster vertex c_i as below:

$$W(c_i) = \sum_{\forall < c_i, v_j > \in E} w(c_i, v_j) + \sum_{\forall < v_j, c_i > \in E} w(c_i, v_j) \quad (2)$$

It has been proven that if we switch the layer assignment of the cluster vertex c_i, the number of via vertices will be reduced by $W(c_i)$. Especially, if $W(c_i)$ is negative then the number of vias will be increased by $|W(c_i)|$; if $W(c_i)$ is zero then the number of vias will not be changed. Therefore, a hill-climbing algorithm is proposed to solve the switching graph problem. The input of the algorithm is a LAP graph, and the output of the algorithm is the optimal switching graph of the LAP graph. The basic idea behind this algorithm is: firstly, we select a cluster vertex c_i in CL such that $W(c_i)$ is the maximum of all cluster vertice. If $W(c_i)$ is greater than zero, then a switching graph is generated by reversing the direction of all the arcs that are incident to c_i. The generated switching graph corresponds to a feasible layer assignmnet of the original constrained via minimization problem, but the number of the vias introduced in this layer assignment is $W(c_i)$ less than that of the previous layer assignment. This procedure is called "selecting-and-switching". The procedure "selecting-and-switching" is repeated until $W(c_i)$ is less than zero. In this manner, a satisfactory layer assignment can be obtained.

Like other hill-climbing algorithms, however, this algorithm cannot guarantee to find an optimal solution as the procedure could terminate at a local minimum in contrast to a global minimum. Thus, in order to increase the possibility of finding the optimal solution, some strategies to escape from a local minimum is adopted. One of the strategies is that we allow the cluster vertex whose $W(c_i)$ is zero to be selected during the course of "selecting-and switching". Although it will not reduce the number of vias immediately, it will benefit to escape from a local minimum. Another strategy to escape from a local minimum is called "mutation". The mutation is to randomly select several cluster vertice to switch. It could lead to increase the number of vias temporally, but it contributes to find a global minimum. Experimental results have shown that these strategies are effective to escape from local minimum and therefore achieve an optimal solution. The algorithm is described as follows:

1. Select a cluster vertex such that its $W(c_i)$ is maximum;
2. If $W(c_i)$ is not negative, then reverse the direction of the arcs that are incident to c_i and go to 1; otherwise,
3. If the result is satisfactory, then exit; otherwise,
4. Randomly select several cluster vertice and reverse the direction of the arcs being incident to these cluster vertice; then, go to 1.

5. Practical Considerations

Due to the fabrication technology, or performance considerations, some practical constraints

TABLE 1. Experimental Results

Examples	# of nets	# of vias before optimization	# of vias after optimization	The rate of reduction (%)	CPU (seconds)	Optimality (yes/no)
Ex. 16(a) in [9]	10	24	4	83.33	0.03	yes
Ex. 17(a) in [9]	18	14	6	57.14	0.01	yes
Ex. 2 in [8]	7	3	1	66.67	0.01	yes

should be taken into account within the constrained via minimization. The constraints are different from a routing environment to another subject to fabrication technology, design requirements and performance considerations. The proposed algorithm can be easily adjusted to meet different practical constraints. Here are some examples:

5.1. The layer assignment of terminals

Generally speaking, the terminals of a routing should be arranged on a particular layer in practical routing. Sometimes, some of the terminals are forced to be arranged on a layer whereas the rest on the other layer. Our algorithm can deal with the situation in the following way: Firstly, for each terminal we check if it is assigned on the expected layer. If not, we flip over the layer assignment of the cluster to which the terminal belongs so that the terminal is assigned to the right layer. Then, we mark all the clusters that contain the terminals and never select those marked cluster vertice to switch. In this way, the terminals can be arranged on the expected layers.

5.2. The layer assignment of special nets

In practical routing, some special nets are required to be assigned onto a particular layer. For example, power lines should always be assigned to metal layer. In line with the technique used for handling terminals' layer assignment problem, we find all the clusters associated to the special nets and adjust the layer assignment of those clusters so that the special nets are assigned on the assigned layer. We mark all the clusters that contain the net segments of the special nets and never select those marked clusters to switch.

6. Experimental Results

The proposed algorithm has been implemented in C++ language on a SunSPARC 20 workstation, and it has been evaluated on some examples. For all the tested examples, optimal layer assignments were obtained. Table 1 shows its performance on the examples. Figure 1(d) is the optimized layer assignment of the initial routing shown in Figure 1(a) obtained by using this proposed approach.

In order to test the efficiency of the proposed algorithm, we further developed a program that can generate randomly an initial routing. And then, we used the implemented program to minimize the number of vias in the initial routing. In this way, we generated and tested six thousands of initial routings. The average sizes of clusters and via candidates were 621 and 428, respectively, and the percentage of via elimination was up to 68% in average.

7. Conclusions

In this paper, we proposed a new approach to the constrained via minimization problem for two-layer routing. This proposed approach can handle both grid-based and gridless routing problems, and allows arbitrary wire segments split at a via candidate. In fact, it can even be used for the routings in which there exist irrational situations that might be raised by designers' interference. Therefore, it is practical. In addition, it is easy to be adjusted to meet various restrictions as we have discussed earlier in this paper. The experimental results have shown that it is effective and efficient.

However, like other constrained via minimization approaches, this approach is not able to eliminate the vias introduced at intrinsic via candidates.

References

[1] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apparatus," *Proc. 8th Design Automation Workshop*, pp. 155-163, 1971.

[2] K.R. Stevens and W.M. Vancleemput, "Global via elimination in generalized routing environment," *Proc. Int. Symposium on Circuits and Systems*, pp. 689-692, 1979.

[3] Y. Kajitani, "On via minimization of routing in a 2-layer board," *Proc. of IEEE International Conference On Circuits and Computers*, pp. 295-298, 1980.

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

[5] R.Y. Pinter, "Optimal layer assignment for interconnection," *Proc. Int. Conf. On Circuits and Systems*, pp. 389-401, 1982.

[6] R.W. Chen, Y. Kajitani, and S.P. Chan, "A graph-theoretic via minimization algorithm for two-layer printed circuit boards," *IEEE Trans. on Circuits and Systems*, Vol. CAS-30, No. 5, pp284-299, 1983.

[7] K.C. Chang and H.C. Du, "Efficient algorithm for layer assignment problem," *IEEE Trans. on Computer-Aided Design*, Vol. CAD-6, pp. 67-78, 1987.

[8] Y.S. Kuo, T.C. Chern, and W.K. Shih, "Fast algorithm for optimal layer assignment," *Proc. 25th ACM/IEEE International Conference on Design Automation*, pp554-559, 1988.

[9] X.-M. Xiong and E.S. Kuh, "A unified approach to the via minimization," *IEEE Trans. on Circuits ans Systems*, Vol. 36, No. 2, pp190-2-4, 1989.

[10] F. Barahona, "On via minimization," *IEEE Trans. on Circuits and Systems*, Vol. CAS-37, pp527-530, 1990.

[11] K. Ahn and S. Sahni, "Constrained via minimization," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 12, No. 2, pp273-282, 1993.