

How to Obtain More Compactable Channel Routing Solutions

Jingsheng Cong

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

D. F. Wong

Department of Computer Science
University of Texas at Austin
Austin, Texas 78712

Abstract

In this paper, we present several powerful techniques which allow us to generate efficiently channel routing solutions which are beneficial to further compaction. Our techniques can be applied to straight track compaction as well as to contour routing compaction, and produce very encouraging results. In particular, for Deutsch's Difficult Example, we obtained a straight track routing solution whose area is 7% less than the best known result after straight track compaction. Also, our router generated a routing solution using less area than reported results of all well known routers after contour routing compaction.

1. Introduction

In VLSI layout design, a significant portion of chip area is used for channel routing. There are several grid-based channel routers which can consistently produce channel routing solutions which are at most one or two tracks within optimal solutions [De76, YoKu82, RiFi82, BuPe83, ReSS85]. A further study [De85] showed that the routing solutions of these grid-based router could be compacted to obtain a 15% - 20% area reduction. Both straight track compaction [De85, WoLi86, Ch86] and contour routing compaction [De85, XiKu87, Ro87] have been investigated. From the experimental results of different channel compactors, it was observed that the amount of area reduction is closely related to the grid-based routing solutions we used. For example, as reported in [De85], the same contour channel compaction algorithm was applied to three optimal 19-track grid-based routing solution of Deutsch's Difficult Example. Three different results were obtained. Since all grid-based channel routers were designed to minimize the number of tracks used and do not take the later compaction step into account, it becomes an important problem for channel routing compaction, as raised in [De85] and [XiKu87], to obtain more compactable channel routing solutions. In this paper, we shall present several efficient techniques which allow us to transform grid-based channel routing solution systematically into more compactable routing solutions.

Usually, contour routing compaction may yield more area reduction than straight track compaction. This further reduction on area might be quite helpful in the situation where a small overflow of a channel may cause re-design of the whole layout. However, there are several potential problems with contour routing compaction. First, it makes the routing geometry much more complicated. Also, in the compacted routing solution we may have long wire paths with only minimum separation, which may lead to a poor yield. On the other hand, straight track compaction can give satisfactory area reduction in most of the cases, and the final compacted routing solutions have much simpler routing geometry. In many chip productions, straight tracks are used for routing implementation. Thus, a practical channel routing compactor should be able to perform either straight track compaction or contour routing compaction. In this paper, we show how to generate more compactable routing solutions

either for straight track compaction or for contour routing compaction. In general, the methods presented in this paper can be used as a preprocessing step of channel compaction.

2. Formulation of the Problem

Let S and S' be two grid-based channel routing solutions for the same channel routing problem. We say that S' is *more compactable* than S if S' uses less routing area than S after channel compaction. In grid-based channel routing, we usually assume that there is a grid structure superimposed on the routing area and wires and vias are dimensionless. A typical grid-based routing solution is shown in Fig. 2-1. When we come to channel compaction, we are no longer restricted to the grid structure, and we must also take the dimension of wires and vias into account. Given a grid-based routing solution S , if we require all the wires on the same track in S remains on the same straight track after compaction, as shown in Fig. 2-2, we call the compaction *straight track compaction*. If we allow wires in S to bend arbitrarily (but still complying with design rules) in the compacted solution, as shown in Fig. 2-3, we call the compaction *contour routing compaction*. We want to obtain grid-based channel routing solutions such that after straight track compaction or contour routing compaction, minimum routing area is used.

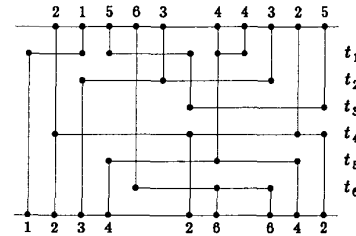


Fig. 2-1 An Example of Two Layer Channel Routing Solution

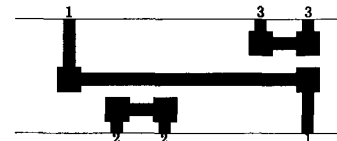


Fig. 2-2 Straight Track Compaction



Fig. 2-3 Contour Routing Compaction

We use the same set of design rules as presented in [De85]:

| | |
|--------------------|------------------|
| path width | 1.0 |
| feature separation | 1.0 |
| size of via | 2.0×2.0 |
| terminal spacing | 4.0 |

It makes comparisons with other compaction results easier by using the same set of design rules. The methods presented in this paper are design rule independent.

In the rest of the paper, we assume that there are two layers available for channel routing. One layer is reserved to horizontal wires and the other layer is reserved to vertical wires. A via is used to connect wires on different layers. Two vias are *adjacent* if they are in the same column on two adjacent tracks and the two vias belong to two different nets. We define the *height* of a channel to be the distance from the bottom edge of the channel to the top edge of the channel.

8. Straight Track Compaction

Straight track compaction is usually achieved by allowing variable track spacing and via offset. If we use uniform track spacing, since there may be two adjacent vias between two tracks, the track spacing should be 3.0 (Fig. 3-1 (a)). However, if we allow variable track spacing, when there is no adjacent vias between the two tracks, we may reduce the track spacing between these two tracks to 2.5 (Fig. 3-1 (b)). Furthermore, if we allow via offset, for example, shifted by 0.1 as in [De85], we may reduce the track spacing between two tracks to 2.8 even if adjacent vias exist between these two tracks (Fig. 3-1 (c)). Clearly, existence of adjacent vias in the grid-based routing solution affects both the channel height and routing geometry after straight track compaction. We call two adjacent tracks with adjacent vias between them a *conflicting track pair*. It is easy to show that for a grid-based routing solution without conflicting track pair, straight track compaction will give a routing solution with minimum channel height using uniform track spacing and without via offset (a much simpler routing geometry!). Thus, the key problem for generating more compactable routing solution for straight track compaction is to minimize the number of conflicting track pairs.

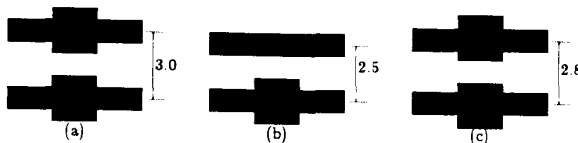


Fig. 3-1 Variable Track Spacing and Via Offset

We introduce two powerful and efficient techniques to transform a grid-based routing solution into another solution with the minimum number of conflicting track pairs. One technique is *track permutation*. For the example shown in Fig. 2-1, there are 4 conflicting track pairs. If we exchange track 2 and 3, we obtain a routing solution with only 2 conflicting track pairs (Fig. 3-2). Another technique we use is *local re-routing*. For the routing solution we obtained in Fig. 3-2, if we reroute net 4 at column 8 and 9,

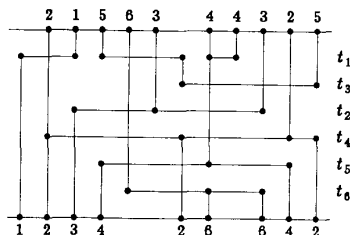


Fig. 3-2 After Exchanging Track 2 and 3 in Fig. 2-1.

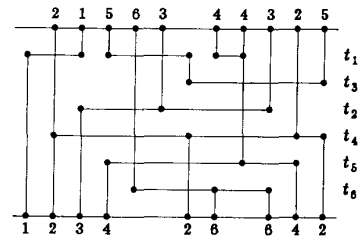


Fig. 3-3 The Solution in Fig. 3-2 after Rerouting.

we can remove adjacent vias between track 5 and 6. Thus, we obtain a routing solution with only one conflicting track pair (Fig. 3-3).

3.1. Track Permutation

We observe that not every track permutation on a routing solution will yield another valid routing solution (i.e. a solution without horizontal and vertical overlap of wires from different nets). To characterize valid track permutations, we define the *track ordering graph* $G(S)$ of a two layer grid-based routing solution S . $G(S)$ is a directed acyclic graph. Each node in $G(S)$ represents a track in S . There is a direct edge from t_i to t_j if at some column, there is a via on t_i above a via of another net on t_j . Fig. 3-4 shows the track ordering graph of the routing example in Fig. 2-1. We can prove that for a two layer routing solution S without unrestricted dogleg, every valid track permutation corresponds to a topological labeling of $G(S)$ [Co87]. The crucial problem left is to obtain a valid track permutation π such that the number of conflicting track pairs in $\pi(S)$ is minimized for a given S , where $\pi(S)$ denotes the two layer solution obtained by permuting the tracks in S according to π . We can show that the problem of finding an optimal track permutation of a given routing solution S is equivalent to the Separation Problem for directed acyclic graphs, which is formulated as follows [LeVW84]:

Input: A given directed acyclic graph G .

Question: Find a topological labeling of G that minimizes the total number of edges between vertices with consecutive labels.

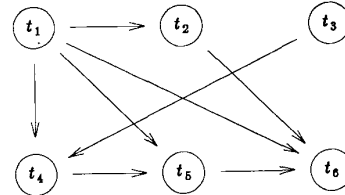


Fig. 3-4 The Track Ordering Graph of the Example in Fig. 2-1.

We leave out the proof for equivalence of the two problems. A linear time algorithm for the Separation Problem was presented in [WoLi86]. By using that algorithm, we can obtain the optimal track permutation in linear time.

3.2. Local Re-routing

Local re-routing is performed after we permute the tracks of the channel routing solution according to the optimal track permutation. By re-routing some nets locally, we can remove some adjacent vias, and, thus, further reduce the number of conflicting track pairs. This task is accomplished by using a maze router, which works as follows: For the partial routing solution in Fig. 3-4 (a), we remove via v_1 (Fig. 3-4 (b)) and try to connect the portion of the net containing x with the portion of the net containing the horizontal segment h_1 (Fig. 3-4 (c)). We try to remove via v_2 in a similar way if the removal of via v_1 fails. Our maze router is based on the classical wave propagation algorithm of Lee [Le61].

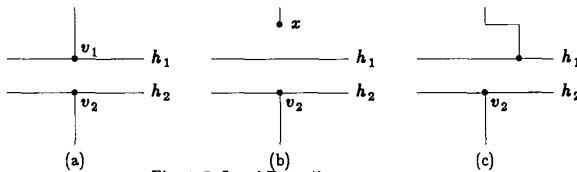


Fig. 3-5 Local Rerouting

In our maze router, we sometimes allow a short vertical wire to be put on the horizontal layer, or a short horizontal wire to be put on the vertical layer. Also, when the original routing solution is very crowded, local re-routing may not be very effective to remove adjacent vias. In this case, we might want to insert an empty track at the most crowded area. Using the empty track for local re-routing again, we might be able to reduce the number of conflicting track pairs significantly. Thus, the compacted channel height may decrease. Our maze router is intelligent enough to decide whether an empty track should be inserted based on whether the final compacted channel height will decrease.

4. Contour Routing Compaction

Contour routing compaction is usually done by processing tracks one by one from the bottom of the channel to the top of the channel. For each track, we assign wires and vias on the track to the lowest possible vertical position. Since we do not have to keep distance between two tracks equal, adjacent vias are no longer the critical consideration. However, the phenomenon shown in Fig. 4-1, referred to as "bump propagation" in [De85] may affect the final compacted result. A via on a track near the bottom of the channel may cause adjacent tracks to jog around it and propagate all the way up to the top. Thus, we want to find routing solutions such that the number of vias on the tracks near the bottom of the channel is minimized. Again, track permutation and local re-routing turn out to be powerful tools to transform standard grid-based routing solutions into the solutions of this kind.



Fig. 4-1 Bump Propagation

4.1. Track Permutation

Assume v_i is the number of vias on track t_i . Let t_i be the i -th track from the top. Then the propagation length caused by all the vias on t_i can be estimated by $i \cdot v_i$, because i measures the distance from track t_i to the top of the channel. Thus, we want to find a valid track permutation π such that $\sum_{i=1}^n \pi(i) \cdot v_i$ is minimized, where $\pi(i)$ is the position of track t_i from the top of the channel after permutation π is performed. Again, valid track permutations can be characterized by topological labelings on the track ordering graph, as defined in the last section. And we can reduce the problem of finding an optimal track permutation to the following single machine job sequencing problem:

Input n jobs of unit execution time with precedence constraints imposed by a given directed acyclic graph to be processed by a single machine. Each job t_i has a weight w_i ($i = 1, 2, \dots, n$).

Question Find a feasible sequence such that the weighted completion time $\sum_{i=1}^n w_i \cdot C_i$ is minimized, where C_i is the completion time of job t_i .

This sequencing problem has been proved to be NP-Complete [La78]. However, we can show that in our case for most of the problems, the number tasks is small (≤ 40), and usually the number of feasible sequencing solutions are quite limited. Thus, branch and

bound algorithm can obtain the optimal result quickly. Also, Sidney's decomposition theorem [Si75] can be used to find the optimal solution.

4.2. Local Re-routing

In this step, we try to re-route some of the nets to remove unnecessary vias by a maze router. This will help to minimize the final compacted channel height, since each via will cause 'bump propagation'. Since we do not want to introduce new vias when we do re-routing, our maze router does not switch layers. Also, since removal of some of the vias may block the removal of other vias, and we want to minimize the number of vias at the bottom of a channel as much as possible because of 'bump propagation', our maze router try to remove vias from the bottom of the channel to the top of the channel. We use a similar wave propagation algorithm as presented in the last section to implement our maze router. Local re-routing is performed after the tracks the channel routing solution is permuted according to the optimal track permutation.

5. Experimental Results

We implemented our algorithms for generating more compactable routing solutions both for straight track compaction and for contour routing compaction respectively. Our programs are written in Pascal language running under Unix 4.3BSD on a Pyramid machine. Table 5-1 shows the routing solutions we obtained for straight track compaction. YK3a, YK3b and YK3c are examples 3a, 3b and 3c, respectively in Yoshimura and Kuh's paper [YoKu82]. D1, D2 and D3 are from the GTE Layout published in [De78]. Diff is the famous Deutsch's Difficult Example. Our router removed all the adjacent vias without inserting empty tracks for all the examples except Deutsch's Difficult Example. Thus, after straight track compaction, we achieved the minimum channel height using uniform spacing. For Deutsch's Difficult Example, we obtained two routing solutions. Both have a smaller channel height than the best reported result (of height 54.8 [De85]) based on the same set of design rules after straight track compaction. The first one has 4 conflicting track pairs, and its channel height is 7% smaller than the best reported result after straight track compaction with variable spacing and via offset. The second one resolved all the conflicting track pairs by inserting an empty track (Fig. 5-1). It has a slightly larger channel height but can be implemented by uniform spacing after straight compaction. It is the first 20 track solution without adjacent vias for Deutsch's Difficult Example ever reported. Running time of our program for straight track compaction on all the tested examples is less than 50 cpu seconds.

| Ex. | d | # of tracks | # of conflicting track pairs | channel height after compaction | lower bound for channel height |
|------|----|-------------|------------------------------|---------------------------------|--------------------------------|
| YK3a | 15 | 15 | 0 | 39 | 39 |
| YK3b | 17 | 17 | 0 | 44 | 44 |
| YK3c | 18 | 18 | 0 | 46.5 | 46.5 |
| D1 | 18 | 18 | 0 | 46.5 | 46.5 |
| D3 | 15 | 15 | 0 | 39 | 39 |
| D5 | 17 | 17 | 0 | 44 | 44 |
| Diff | 19 | 19 | 1 | 50.7 | 49 |
| Diff | 19 | 20 | 0 | 51.5 | 49 |

Table 5-1 Experimental Results for Straight Track Compaction.

Table 5-2 shows the routing result of Deutsch's Difficult Example we obtained for contour routing compaction and comparisons with other results. Our router generated a 19-track grid-based solution, which yields a smaller channel height than all the reported results of well known routers after contour routing compaction. Our

result is the same as the result reported in [De85]. The result in [De85] was based on a partially hand edited 19-track solution, compacted from column 60 to 100. Running time of our program for contour channel routing is less than 12 cup minutes on all the tested examples.

| Based solution | # of tracks | channel height after compaction |
|----------------|-------------|---------------------------------|
| [De76] | 21 | 49.0 |
| [YoKu82] | 20 | 47.0 |
| [RiFi82] | 20 | 48.0 |
| [BuPe83] | 19 | 46.0 |
| [ReSS85] | 19 | 47.0 |
| [De85] | 19 | 45.0 |
| [Ro87] | 81 | 46.0 |
| Ours | 19 | 45.0 |

Table 5-2 Comparisons of Contour Routing Compaction on Deutsch's Difficult Example

6. Remarks and Conclusions

In this paper, we showed how to generate channel routing solutions which are beneficial to later compaction. The basic idea is to do solution transformation based on routing solutions produced by good channel routers. Compaction results based on our solution have not only a smaller channel height but also a simpler routing geometry. Running time for our algorithms is quite short, thus, our program can be used efficiently as a pre-processing step of channel routing compaction to improve compaction results greatly. To the authors' knowledge, this is the first paper explicitly addressing the problem of generating more compactable solutions, although the same problem was raised a few times in literatures [De85, XiKu87]. We would like to see more research along this line.

7. Acknowledgements

We would like to thank Prof. C. L. Liu for his helpful discussion. The first author is partially supported by National Science Foundation under grant MIP 87-03273, by the Semiconductor Research Cooperation under contract 87-DP-109, and by a grant from General Electric Company.

References

[BuPe83] Burstein, M. and R. Pelavin, "Hierarchical Channel Router". *Integration, the VLSI journal* (1983) Vol. 1, pp. 21-38.

[Co87] Cong, J., "A New Approach to Three Layer Channel Routing", MS-Thesis, Dept. of Computer Sci., Univ. of Illinois, Urbana, Illinois, May, 1987.

[De76] Deutsch, D. N., "A Dogleg Channel Router". *Proc. 13th Design Automation Conf.* (1976) pp. 425-433.

[De85] Deutsch, D. N., "Compacted Channel Routing". *ICCAD-85* (Nov., 1985) pp. 223-225.

[La78] Lawler, E. L., "Sequencing Jobs to Minimise Total Weighted Completion Time Subject to Precedence Constraints". *Annals of Discrete Mathematics* (1978) Vol. 2, pp. 75-90.

[Le61] Lee, C. Y., "An Algorithm for Path Connection and its Application". *IRE Trans. on Electronic Computers* (1961) Vol. EC-10, pp. 346-365.

[LeVW84] Leung, J., O. Vornberger and J. D. Witthoff, "On some variants of the bandwidth minimisation problem". *SIAM J. Comput.* (1984) Vol. 13, No. 3, pp. 650-687.

[ReSS85] Reed, J., A. Sangiovanni-Vincentelli and M. Santomauro, "A New Symbolic Channel Router: YACR2". *IEEE Trans. on Computer Aided Design of CIAS* (1985) Vol. CAD-4, No. 3, pp. 208-219.

[RiFi82] Rivest, R. L. and C. M. Fiduccia, "A 'Greedy' Channel Router". *Proc. 19th Design Automation Conf.* (1982) pp. 418-424.

[Ro87] Royle, J. et al, "Geometrical Compaction in One Dimension for Channel Routing". *Proc. of 24th Design Automation Conf.* (June, 1987) pp. 140-145.

[Si85] Sidney, J. B., "Decomposition Algorithms for Single-Machine Sequencing with Precedence Relations and Deferral Costs". *Operations Research* (1975) Vol. 23, No. 2, pp. 283-298.

[WoLi86] Wong, D. F. and C. L. Liu, "Compacted Channel Routing with Via Placement Restriction". *Integration, the VLSI journal* (1986) Vol. 4, pp. 267-307.

[XiKu87] Xiong, X. and E. S. Kuh, "Nutcracker: An Efficient and Intelligent Channel Spacer". *Proc. of 24th Design Automation Conf.* (1987) pp. 298-304.

[YoKu82] Yoshimura, T. and E. S. Kuh, "Efficient Algorithms for Channel Routing". *IEEE Trans. on Computer Aided Design of ICAS* (Jan. 1982) Vol. CAD-1, pp. 25-35.

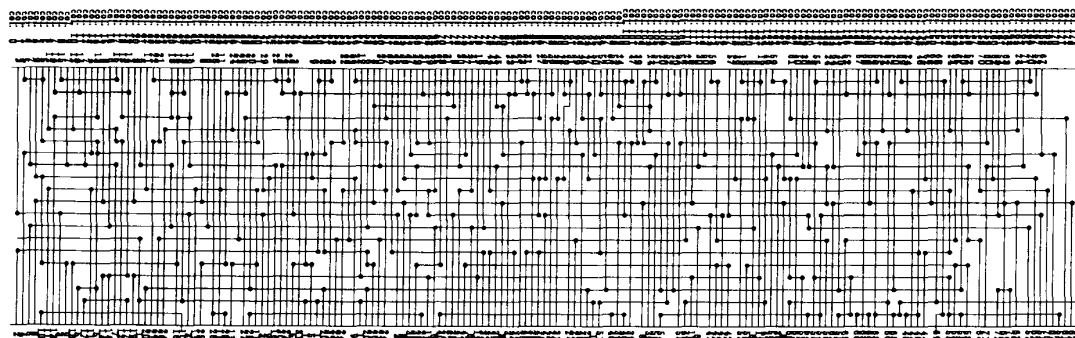


Fig. 5-1 A 20-straight-track solution without adjacent vias of Deutsch's Difficult Example.