ELSEVIER

INTEGRATION, the VLSI journal 17 (1994) 141–151

INTEGRATION
the VLSI journal

Single phase three-layer channel routing algorithms☆

Teofilo F. Gonzalez[a,*], Si-Qing Zheng[b]

[a] *Department of Computer Science, University of California, Santa Barbara, CA 93106, USA*
[b] *Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA*

Received 1 November 1993

Abstract

We present a simple three-layer assignment algorithm for planar layouts generated by a class of layout algorithms. This class of algorithms includes simple variations of the following algorithms developed by Mehlhorn et al. (1986):
(i) The algorithm that generates planar layouts for the two-terminal net CRP with d_{max} tracks.
(ii) The algorithm that generates planar layouts for the two- and three-terminal net CRP with at most $\lfloor 3d_{max}/2 \rfloor$ tracks.
(iii) The algorithm that generates planar layouts for the multiterminal net CRP with at most $2d_{max} - 1$ tracks.

The planar layouts generated by these algorithms are three-layer wirable by the layer assignment algorithm given in Preparata and Lipski (1984). Our approach is different. We make slight modifications to these layout algorithms and incorporate a simple layer assignment strategy to generate three-layer wirings. Our algorithms are faster and conceptually simpler because there is no need to construct diagonal diagrams and legal partitions. The channel width of the wiring generated by our algorithm is identical to that of the corresponding planar layout generated by algorithms (i)–(iii). Our layer assignment methodology can also be used to develop other single-phase three-layer algorithms, as demonstrated by Wieners–Lummer (1991).

Keywords: VLSI design automation; Three-layer wiring; Knock-knee layout; Channel routing

1. Introduction

The channel routing problem (CRP) has been recognized as a fundamental problem in computer-aided VLSI layout design. The CRP is defined over the rectangular grid formed by lines $\{x = i \mid i \in Z\}$ and $\{y = j \mid 0 \leqslant j \leqslant h + 1 \text{ and } j \in Z\}$. The horizontal grid lines $y = 0$ and $y = h + 1$ are called the *boundaries* of the channel, and the horizontal grid lines $y = j$, $1 \leqslant j \leqslant h$, are called *tracks* of the channel. All the vertical grid lines are called *columns* of the channel. A channel routing problem consists of a collection of pairwise disjoint sets of grid points, $N = \{N_1, N_2, \ldots, N_m\}$,

*Corresponding author.
☆This research was supported in part by the National Science Foundation under Grant DCR-8503163.

SSDI 0167-9260(94)00008-5

located on the channel boundaries. Each N_i in N is called a *net* and each point in N_i is called a *terminal* of net N_i. Terminals in each net need to be connected by wires running along the grid lines. It should be noted that the terms "layout" and "wiring" are frequently interchangeable. In this paper we follow Preparata and Lipski's [1] convention and give these two terms a different connotation. A *planar layout* (or simply a layout) of a channel routing problem is a collection of edge disjoint connected subgraphs $W = \{W_1, W_2, \ldots, W_m\}$ of the channel grid, such that each W_i connects all the terminals in net N_i. This definition implies that at each grid point of the channel there can be at most two wires; furthermore, for $i \neq j$, W_i and W_j do not share any grid line segment. There are several different wiring models for the channel routing problem. In this paper, we consider the CRP under the knock-knee model. In this model, when two wires share a grid point they either *cross* or form a *knock-knee*. The two types of knock-knees are given in Fig. 1. The horizontal (vertical) portions of a knock-knee are called the *horizontal (vertical) arms of the knock-knee*.

Assume that there are $k \geqslant 2$ conducting layers $L_1, L_2, \ldots, L_k$ available, and L_{i+1} is stacked on top of L_i, for $1 \leqslant i < k$. A *wiring* for layout $W = \{W_1, W_2, \ldots, W_m\}$ is a mapping that associates each edge of W_i, $1 \leqslant i \leqslant n$, to a layer in such a way that for every $i \neq j$ if edges (p_1, p_2) and (p_2, p_3) in W_i are assigned to L_s and L_t, respectively, and edge (p_2, p_4) in W_j is assigned to L_u, then either $u > \max\{s, t\}$ or $u < \min\{s, t\}$. Contact cuts (called *vias*) can be established only at grid points. Vias allow a wire to change from one layer to the another. The premier objective of the channel routing problem consists of finding an *optimal wiring*, i.e., a wiring on a grid with least number of horizontal grid lines. We also refer to this criteria as *minimum channel width* or minimum number of tracks. Other optimization objectives include minimize the area of the smallest rectangle enclosing the wiring, minimize the number of layers, and minimize the total number of vias. We call the open interval $(c, c + 1)$ a *vertical cut*, where c and $c + 1$ are two adjacent columns of the channel. We define the *channel density* $d_{\max}$ for a CRP as $d_{\max} = \max\{d(c)\}$, where $d(c)$, the *local density* of the vertical cut $(c, c + 1)$, is the number of nets in N whose leftmost terminal and rightmost terminal are located to the left and right of vertical cut $(c, c + 1)$, respectively. Since every net that contributes to $d_{\max}$ must have a horizontal wire along the same vertical cut, we know that $d_{\max}$ is a lower bound for the channel width of an optimal wiring.

Constructing a multilayer wiring with the least number of layers for any given layout W is a fundamental problem in CAD of VLSI. Whether or not a given planar layout is one-layer or two-layer wirable can be easily determined. To construct a three-layer wiring for a planar layout, a powerful transformation, legal partition of the diagonal diagram induced by the layout, was

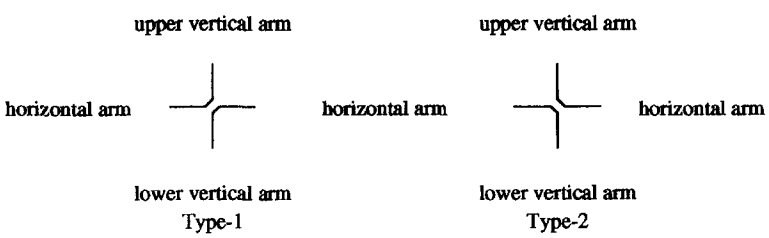


Fig. 1. Knock-knees.

introduced in [1, 2]. Based on this transformation, Lipski [3] showed that there are planar layouts that are not three-layer wirable. Gonzalez and Zheng [4] showed that there even exist six-row planar layouts which are not three-layer wirable. In general, the problem of determining whether a given planar layout is three-layer wirable is NP-complete [3]. Brady and Brown [5] showed that every planar layout is four-layer wirable by finding a legal partition (that satisfies some additional properties) of the diagonal diagram induced by the layout. An alternative layer assignment algorithm is proposed in [6]. This algorithm attempts to construct a three-layer wiring first, and if it fails, a four-layer wiring is generated. There is no known characterization of the class of planar layouts for which this algorithm generates three-layer wirings.

The general multilayer wiring theory suggests a two-phase approach to the CRP problem. In the first phase, a planar layout is generated; then, in the second phase, a multilayer wiring of the layout is constructed using the layer assignment methods of [2, 5, 6]. Indeed, most existing knock-knee mode CRP algorithms follow this approach. These two-phase CRP algorithms include the ones in [1, 7–11]. The two-terminal net CRP algorithm given in [1, 9] generates optimal three-layer wirings. However, the one in [9] is simpler. Recently, Kuchem et al. developed a two-phase three-layer algorithm for the two-terminal net CRP. Their algorithm guarantees a wiring solution with minimum channel width and near-optimal area [11]. For the case when each net consists of at most three terminals, the algorithms in [9, 10] generate three-layer wirings with no more than $\lfloor 3d_{\max}/2 \rfloor$ tracks. For multiterminal net channel routing problems, the algorithms in [12, 9, 8], generate three-layer wirings with at most $2d_{\max} - 1$ tracks. Gao and Kaufmann [7] showed that a planar layout with at most $3d_{\max}/2 + O(\sqrt{d_{\max} \log d_{\max}})$ tracks for any multiterminal net CRP can be constructed efficiently. These layouts are in general not three-layer wirable [13]. Recently, Wieners-Lummer [13] designed a single-phase three-layer routing algorithm for the multiterminal net CRP. The approach, based on the algorithm of [7], is to decompose a CRP into three subproblems, and solve each by an algorithm similar to the conservative layout algorithms discussed in Section 2. The layer assignment rules are based on our rules, which were reported in a preliminary version of this paper [14]. Wieners–Lummer's algorithm guarantees three-layer wirings with channel width no more than $3d_{\max}/2 + O(\sqrt{d_{\max} \log d_{\max}})$.

In this paper we present a simple three-layer assignment algorithm for planar layouts generated by a class of layout algorithms. This class includes the following algorithms developed by Mehlhorn et al. [9].

(i) The algorithm that generates planar layouts for the two-terminal net CRP with $d_{\max}$ tracks.
(ii) The algorithm that generates planar layouts for the two- and three-terminal net CRP with at most $\lfloor 3d_{\max}/2 \rfloor$ tracks.
(iii) The algorithm that generates planar layouts for the multiterminal net CRP with at most $2d_{\max} - 1$ tracks.

The planar layouts generated by these algorithms are three-layer wirable by the layer assignment algorithm given in [1]. Our approach is different. We make slight modifications to these layout algorithms and incorporate a simple layer assignment strategy to generate three-layer wirings. Our algorithms are faster and conceptually simpler because there is no need to construct diagonal diagrams and legal partitions. The channel width of the wiring generated by our algorithm is identical to that of the corresponding planar layout generated by algorithms (i)–(iii). Our layer assignment methodology can also be used to develop other single-phase three-layer algorithms, as demonstrated in [13].

2. Three-layer wiring algorithms

Let A be a layout algorithm that generates a layout by a single left-to-right column-by-column sweeping (scanning) of terminals. For column c, we define the *strip area* $S(c)$ *around column* c as the area delimited by the two vertical lines $c - 1/2$ and $c + 1/2$, and the top and bottom channel boundaries. When algorithm A considers column c, it generates the layout $W(c)$ for the strip area $S(c)$. The horizontal wires leaving $W(c)$ from the right of $S(c)$ are called *the output wires of* $W(c)$. Assume that the leftmost terminal of CRP is in column 1. For $c > 1$, *the input wires* of $W(c)$ are the output wires of $W(c-1)$, and there are no input wires for $W(1)$. A horizontal wire that is both an input wire and an output wire on the same track in $W(c)$ is called a *continuing wire*. A horizontal wire that is only an output wire on some track in $W(c)$ is called a *beginning wire*. We say that a wire W_i is a *k-stranded input (output) wire in* $W(c)$ if the vertical line $x = c - 1/2$ ($x = c + 1/2$) intersects k times wire W_i in layout $W(c)$. We say that vector $V = (v_1, v_2, \ldots, v_m)$ is an *input (output) strand vector of* $W(c)$ if wire W_i is a v_i-stranded input (output) wire in $W(c)$, for $1 \leqslant i \leqslant m$. We use $isv(W(c))$ ($osv(W(c))$) to denote the input (output) strand vector of $W(c)$.

The planar layout algorithms which can be modified by our strategy to generate three-layer wirings are called *conservative planar layout algorithms*. An algorithm A is said to be a conservative layout algorithm if and only if it satisfies the following three properties.

(1) Algorithm A generates a layout by a single left-to-right column-by-column sweep (scan). When column c is being considered the algorithm generates its layout $W(c)$, and once $W(c)$ is generated, the layouts $W(1), W(2), \ldots, W(c)$ will not be modified.

(2) For column $c > 1$ every layout $W'(c-1)$ such that $osv(W'(c-1)) = osv(W(c-1))$ (remember that $W(c)$ is the layout generated by algorithm A for column c), algorithm A generates a layout $W'(c)$ with $osv(W'(c)) = osv(W(c))$.

(3) For any layout $W'(c-1)$ with $osv(W'(c-1)) = osv(W(c-1))$ algorithm A generates a layout $W'(c)$ with no more than two knock-knees. If there are two knock-knees, the knock-knees are of different types, and the type-1 knock-knee is below the type-2 knock-knee. A type-1 (type-2) knock-knee has its lower (upper) vertical arm intersect the bottom (top) boundary.

Without loss of generality, assume that every conservative algorithm is initially assigned h empty tracks and throughout the execution of the algorithm the value h is never increased nor decreased. An algorithm A that does not satisfy this restriction can be easily simulated by executing A once to obtain the value of h. Once this value is computed, algorithm A can be easily modified to satisfy the additional property. Not all of the algorithms (i)–(iii) in [9] satisfy properties (1)–(3). However, by making slight modifications these algorithms can be transformed into conservative layout algorithms without sacrificing performance.

Let A be any algorithm that satisfies properties (1)–(3). In what follows, we define our algorithm, A^*, that constructs a layout (similar to the one constructed by A) and finds its layer assignment simultaneously. Algorithm A^* constructs the layout as algorithm A in a single left-to-right scan of the columns. When column c is being considered, we first take $W^*(c-1)$, the layout generated by algorithm A^* at the $(c-1)$th iteration (if $c = 1$, $W^*(0) = \emptyset$) and mimic algorithm A on this input. Let $W'(c)$ be the layout obtained by this process. Note that the input wires in $W'(c)$ are identical to the output wires in $W^*(c-1)$. Depending on the knock-knees in $W'(c)$ and the layer assignment for $W^*(c-1)$, $W^*(c)$ is defined as either $W'(c)$ or a slightly modified version of $W'(c)$. In either case the output strand vector for $W^*(c)$ and $W'(c)$ are identical. Let $W(1), W(2), \ldots$ (resp. $W^*(1)$,

$W^*(2), \ldots)$ be the layout constructed by algorithm A (resp. A^*) for some CRP problem instance N. From this brief description and the assumption that algorithm A satisfies property (2) one can easily prove that at each step, the output strand vector for $W(c)$ is identical to the output strand vector for $W^*(c)$. When algorithm A^* is processing column c, the layer assignment for each wire segment in $W^*(c)$ is determined. The layer assignment rules are quite simple: horizontal wires are always assigned to either the top layer or the bottom layer, whereas the vertical wires are assigned to the middle layer in "normal" regions and to either the top or bottom layer in other regions. Vias are introduced whenever necessary. For column c, we use $[k_1, k_2]$, where $k_1 \leqslant k_2$, to represent the vertical grid segments from track k_1 to track k_2. Note that it is a closed interval. For open intervals we use parentheses instead of square brackets. Remember that the bottom (top) boundary is track 0 ($h + 1$). We define the *strip area* $S(c,I)$ ($S(c)$ *restricted to* I) as the set of all points in $S(c)$ with y-coordinate value $y \in I$, where $I = [k_1, k_2]$ for some $k_1 < k_2$. Similarly, the layout $W^*(c,I)$ ($W'(c,I)$) is defined as $W^*(c)$ ($W'(c)$) restricted to the strip area $S(c,I)$. Depending on the knock-knees in $W'(c)$, each of the vertical grid segments in column c is labeled, R_1, R_N, or R_2. A region R_N is a normal region, and the other two regions, R_i, contain exactly one type-i knock-knee. The labeling procedure is given below.

procedure LABELING

case:
:there is no knock-knee in $W'(c)$ /* Fig. 2(a) */:
the interval $[0, h + 1]$ is labeled R_N;
:there is exactly one knock-knee in $W'(c)$ and it is type-1 /* Fig. 2(b) */:
let k be the track where the knock-knee is located;
the interval $[0, k - 1)$ is labeled R_N, $[k - 1, k]$ is labeled R_1, and $(k, h + 1]$ is labeled R_N.
:there is exactly one knock-knee in $W'(c)$ and it is type-2 /* Fig. 2(c) */:
let k be the track where the knock-knee is located;
the interval $[0, k)$ is labeled R_N, $[k, k + 1]$ is labeled R_2, and $(k + 1, h + 1]$ is labeled R_N.
:there are two knock-knees in $W'(c)$ /* Figure 2(d) */:
/* By property (3) we know that the knock-knees are of different types and that the type-1 knock-knee is below the type-2 knock-knee */
let $k_1(k_2)$ be the track where the type-1 (type-2) knock-knee is located;
/* By property (3) we know that $k_1 < k_2$ */
the interval $[0, k_1 - 1)$ is labeled R_N, $[k_1 - 1, k_1]$ is labeled R_1, (k_1, k_2) is labeled R_N, $[k_2, k_2 + 1]$ is labeled R_2, and $(k_2 + 1, h + 1]$ is labeled R_N.
endcase

end of procedure LABELING

In the following figures the region enclosed by wiggled lines is a three-layer wiring, and the region not enclosed by wiggled lines is a planar layout. In a three-layer wiring we use solid lines to represent wires assigned to the top layer, dashed lines for wires assigned to the bottom layer and dotted lines for the wires assigned to the middle layer. Let us now explain how the layout $W^*(c)$ and a layer assignment for it is generated by algorithm A^*. Algorithm A^* generates $W^*(c)$ by

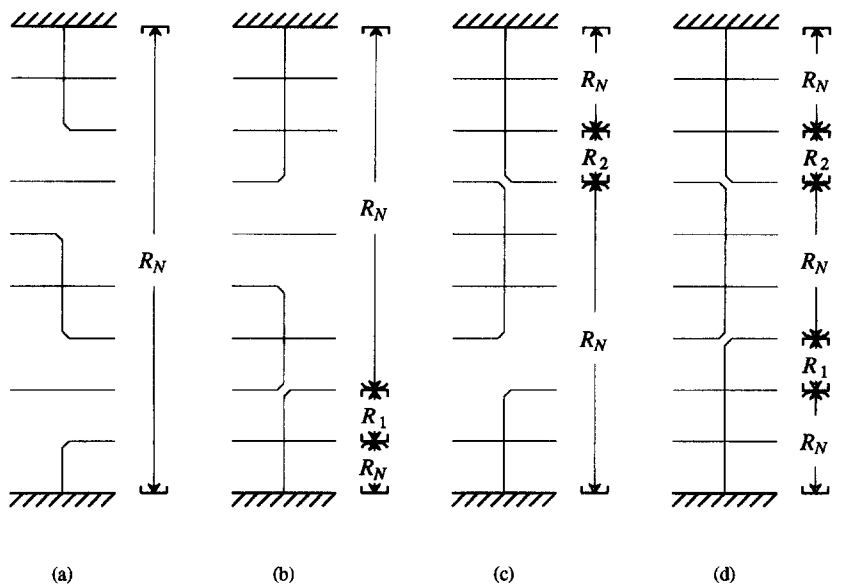


Fig. 2. Labeling examples.

constructing sublayouts $W^*(c, I)$ for each uniformly labeled interval I in $W'(c)$. For the uniformly labeled interval I, we set initially the input wires in $W^*(c, I)$ to be the same as the output wires in $W^*(c-1, I)$ which are identical to the set of input wires in $W'(c, I)$. The layer assignment for the input wires in $W^*(c, I)$ is identical to the layer assignment for the output wires in $W^*(c-1, I)$. Depending on the label assigned to an interval I, the remaining part of the layout, $W^*(c, I)$, and its layer assignment is constructed as follows.

procedure LAYOUT_AND_LAYER_ASSIGNMENT

Case 1: Interval I is labeled R_N. Algorithm A^* generates the layout $W^*(c, I) = W'(c, I)$. The layer assignment is defined as follows. Remember that the input wires in $W^*(c, I)$ are assigned to the same layers as the output wires in $W^*(c-1, I)$. All vertical wires are assigned to the middle layer, all the continuing wires remain in the layer assigned to their input portion, and the beginning wires are assigned to the top layer (note that they could have also been assigned to the bottom layer) (see Fig. 3).

Case 2: Interval I is labeled R_1. Clearly, the interval is of the form $I = [k-1, k]$ for some track k, there is exactly one type-1 knock-knee in $W'(c, I)$ and it is located at grid point (c, k). There are two cases.

Subcase 2.1: There is no input wire in $W'(c, I)$ assigned to track $k-1$, or the input wire in $W'(c, I)$ assigned to track $k-1$ is electrically common with the lower vertical arm of the knock-knee.

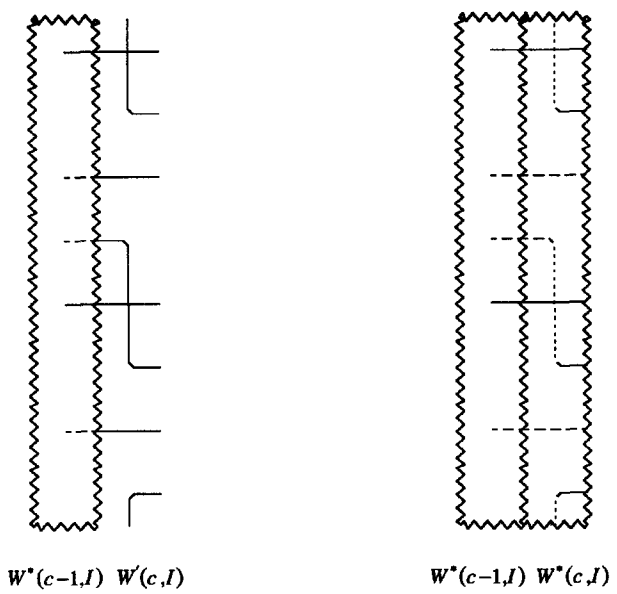


Fig. 3. Example for case 1.

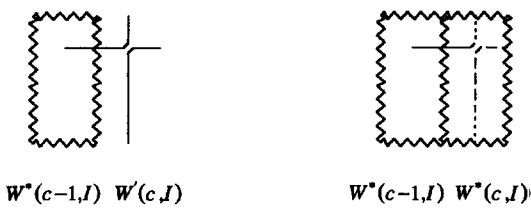


Fig. 4. Example for case 2.1.

Algorithm A^* generates the layout $W^*(c,I) = W'(c,I)$. The layer assignment is defined as follows. If the input wire in track k of $W^*(c,I)$ is in the top (bottom) layer, then the only vertical wire and all the output wires in $W^*(c,I)$ are assigned to the bottom (top) layer (see Fig. 4).

Subcase 2.2: The input wire in $W'(c,I)$ assigned to track $k-1$ is not electrically common with the lower arm of the knock-knee.

There are two cases depending on whether the output wires in $W^*(c-1,I)$ assigned to tracks $k-1$ and k are in the same layer or not.

Subcase 2.2.1: The output wires in $W^*(c-1,I)$ assigned to tracks $k-1$ and k are in the same layer.

Algorithm A^* generates the layout $W^*(c,I) = W'(c,I)$. The layer assignment is defined as follows. If the output wires in tracks $k-1$ and k of $W^*(c-1,I)$ are in the top (bottom) layer, then the only vertical wire and the beginning horizontal wires in $W^*(c,I)$ are assigned to the bottom (top) layer, and the continuing wire in $W^*(c,I)$ remains in the same layer as its input portion (see Fig. 5).

Subcase 2.2.2: The output wires assigned to tracks $k-1$ and k in $W^*(c-1, I)$ are in different layers.

In this case the layout $W^*(c, I) \neq W'(c, I)$. Layout $W^*(c, I)$ is $W'(c, I)$ after performing the transformation shown in Fig. 6.

The layer assignment is defined as follows. If the output wire in track k of $W^*(c-1, I)$ is in the top (bottom) layer, then the only vertical wire and the output wire in track k of $W^*(c, I)$ are assigned to the bottom (top) layer. The output wire in track $k-1$ of $W^*(c, I)$ is assigned to the top (bottom) layer (see Fig. 7).

Case 3: Interval I is labeled R_2. We omit this case since it is symmetric to case 2.

end of procedure LAYOUT_AND_LAYER_ASSIGNMENT

In Fig. 8 we give a planar layout constructed by algorithm (i) in [9]. The corresponding layout constructed by our procedure is given in Fig. 9.

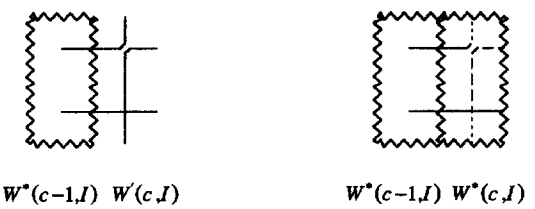


Fig. 5. Example for case 2.2.1.

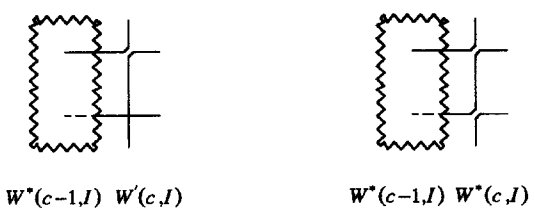


Fig. 6. Layout transformation for case 2.2.2.

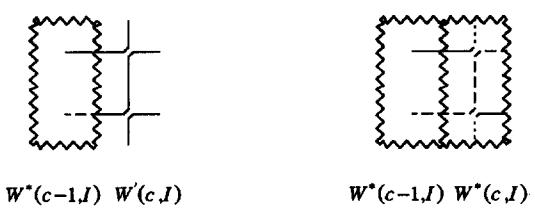


Fig. 7. Example for case 2.2.2.

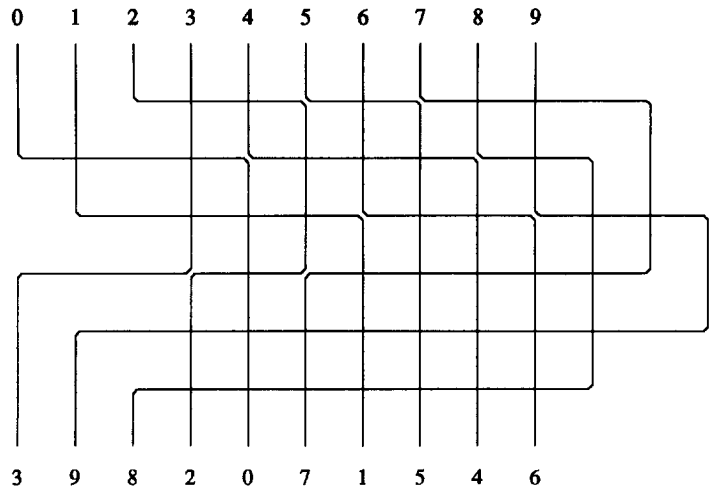


Fig. 8. Layout constructed by algorithm (i) in [9].

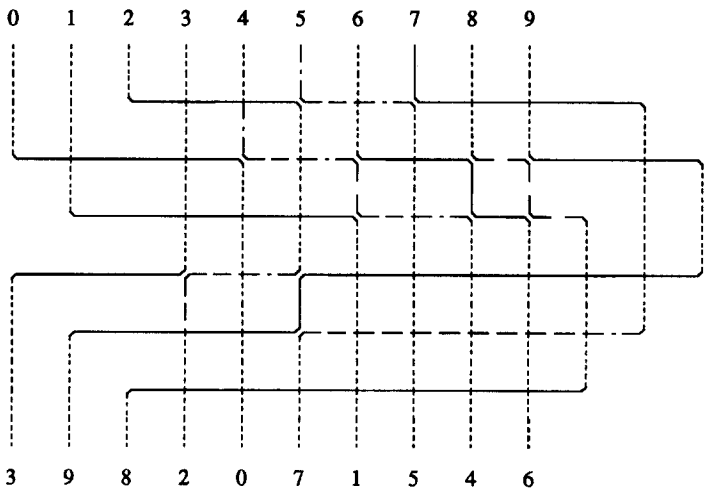


Fig. 9. Layout constructed by our procedure.

Theorem 2.1. *Let A and A* be defined as above. For a set N of nets, algorithm A* constructs a planar layout and wires it in three layers. Furthermore, the number of tracks in the three-layer wiring constructed by A* and the number of tracks in the planar layout constructed by A are identical.*

Proof: Since algorithm A satisfies property (1) we know that it constructs the planar layout in a single left-to-right scan of the columns. This implies that once the layout for $W(c)$ is constructed, the layouts for $W(1), W(2), \ldots, W(c)$ are not changed. Algorithm A^* mimics this process. If $isv(W^*(c)) = isv(W'(c))$, then we know by property (2) and our construction rules that the $osv(W^*(c)) = osv(W'(c))$. Therefore, it follows inductively (after a trivial proof for the base) that

$W^*(c)$ and $W(c)$ have identical output strand vectors in each column c. One can easily verify that W^* is a planar layout for N, and that the three-layer wiring generated by our procedure is valid. Therefore, algorithm A^* constructs for N a planar layout and wires it in three layers; furthermore, the number of tracks in the three-layer wiring constructed by A^* and the number of tracks in the planar layout constructed by A are identical. This completes the proof of the theorem. □

3. Discussion

There are two major approaches for solving three-layer routing problems: the two-phase approach and the single-phase approach. In the two-phase approach, a planar layout is constructed in the first phase. In the second phase a three-layer wiring for the planar layout obtained in the first phase is constructed through a transformation, e.g., legal partition of the diagonal diagram induced by the layout. In the single-phase approach, layout construction and the three-layer assignment of the layout are performed simultaneously.

In this paper we presented a simple method to convert two-phase three-layer conservative routing algorithms into single-phase three-layer routing algorithms. This class of algorithms includes simple variations of well-known algorithms for the channel routing problem. Our approach makes simple modifications to conservative layout algorithms and incorporates a simple layer assignment strategy. We believe that if the structure of the planar layouts generated by a layout algorithm are simple, a three-wiring for the layout may be found by using diagonal diagrams. On the other hand, if a layout algorithm generates planar layouts with simple structures, it is not unlikely that this layout algorithm can be transformed into a single-phase routing algorithm. There is a broader class of algorithms for which transformations similar to ours can generate three-layer wirings in a single phase. Property (2) defined in this paper is too restrictive. We defined conservative algorithms this way in order to simplify our proof. One may relax property (2) and only require that the number of extended nets, paired nets, etc., have identical counts at the end of each step. Equivalence proofs can also be obtained for these cases. For brevity we did not include the broader class of layout algorithms.

In general, wirings generated through legal partitions of the diagonal diagrams tend to have a large number of vias. For the layouts whose diagonal diagrams satisfy certain properties, some techniques can be used to reduce the number of vias. For example, the layouts generated by the three algorithms given in [9] can be wired in three layers by using the layer assignment algorithm given in [1]. This layer assignment algorithm finds a legal partition of the diagonal diagram corresponding to the layout. Special techniques are used to minimize the number of vias in the three layer wiring. It is easy to show that our layer assignment algorithm has similar performance with respect to the number of vias.

Recently, Wieners-Lummer [13] designed a single-phase three-layer routing algorithm for the multiterminal net CRP. The approach, based on the algorithm of [7], is to decompose a CRP into three subproblems, and solve each by an algorithm similar to the conservative layout algorithms discussed in Section 2. The layer assignment rules are based on our rules, which were reported in a preliminary version of this paper [4]. Wieners-Lummer's algorithm guarantees three-layer wirings with channel width no more than $3d_{\max}/2 + O(\sqrt{d_{\max} \log d_{\max}})$.

References

[1] F.P. Preparata and W. Lipski, Jr., Optimal three-layer channel routing, *IEEE Trans. Comput.* **33**(5) (1984) 427–437.

[2] W. Lipski, Jr. and F.P. Preparata, A unified approach to layout wirability, *Math. Systems* **19** (1987) 189–203.

[3] W. Lipski, Jr., An NP-complete problem related to three-layer channel routing, *Adv. Comput. Res.* **2** (1984) 231–244.

[4] T. Gonzalez and S.-Q. Zheng, Three-layer wirability of planar layouts, *Proc. 25th Allerton Conf. on Communication, Control and Computing*, Urbana-Champaign, Illinois (1987) 387–396.

[5] M.L. Brady and D.J. Brown, VLSI routing: four layers suffice, *Adv. Comput. Res.* **2** (1984) 245–258.

[6] I.G. Tollis, A new approach to wiring layouts, *IEEE Trans. Comput. Aided Des. Integrated Circuits Systems* **10**(11) (1991) 1392–1400.

[7] S. Gao and M. Kaufmann, Channel routing of multiterminal nets, *Proc. 28th Symp. on Foundations of Computer Science* (1987) 316–325.

[8] T. Gonzalez and S.-Q. Zheng, Three-layer channel routing of multi-terminal nets, Technical Report, #87–13, CS Dept., UC Santa Barbara, August 1987.

[9] K. Mehlhorn, F.P. Preparata and M. Sarrafzadeh, Channel routing in knock-knee mode: simplified algorithms and proofs, *Algorithmica* **1**(2) (1986) 213–221.

[10] F.P. Preparata and M. Sarrafzadeh, Channel routing of nets bounded degree, *VLSI: Algorithms Architectures* (1984).

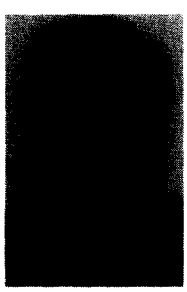
[11] R. Kuchem, D. Wagner and F. Wagner, Area-optimal three-layer channel routing, *Proc 30th IEEE Symp. on Foundations of Computer Science* (1989).

[12] M. Sarrafzadeh and F.P. Preparata, Compact channel routing of multiterminal nets, *Ann. Discrete Math.* **25** (1985) 255–279.

[13] C. Wieners-Lummer, Three-layer channel routing in knock-knee mode, unpublished manuscript, University of Paderborn, Germany, 1991.

[14] T. Gonzalez and S.-Q. Zheng, Simple three-layer channel routing algorithms, *Proc. AWOC' 88, LNCS 319* (Springer, Berlin, 1988) 237–244.

[15] A. Frank, Disjoint paths in a rectilinear grid, *Combinatorica* **2**(4) (1982) 361–371.



Dr. Gonzalez received the B.Sc. degree in Computer Science from the Instituto Technologico de Monterrey (1972) and the Ph.D. degree in Computer Science from the University of Minnesota, Minneapolis, (1975).

Since then Dr. Gonzalez has been a Computer Science faculty member at The University of Oklahoma, The Pennsylvania State University, The Instituto Technologico de Monterrey, The University of Texas at Dallas, and The University of California, Santa Barbara, where he is currently Professor of Computer Science, Professor Gonzalez research activity has concentrated on the development of efficient exact and approximation algorithms for problems in several areas, including CAD/VLSI placement and routing, job scheduling, graph applications, and computational geometry.

Dr. Gonzalez current research interests are: the design of efficient algorithms for component placement and wire routing problems with applications in CAD; algorithms for graph partitioning, grouping and covering problems; and exact algorithms for generalized dictionaries and related problems.

Dr. Gonzalez is an Editor of IEEE Transactions on Computers, and is a member of the Association for Computing Machinery, a full member of the Operations Research Society of America, and an affiliate member of IEEE Computer Society.

Si-Qing Zheng is an Associate Professor of Computer Science at Louisiana State University, Baton Rouge. He received the B.S. degree from Jilin University, China, in 1973. He received the M.S. degree in mathematical sciences from the University of Texas at Dallas in 1982, and the Ph.D. degree in computer science from the University of California at Santa Barbara in 1987. Dr. Zheng's research interests include VLSI, parallel and distributed computing, and computational geometry.