Simple Three-Layer Channel Routing Algorithms ‡

Teofilo Gonzalez and Si-Qing Zheng‡‡
University of California, Santa Barbara

Abstract. In this paper 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 currently best algorithms for the three layer channel routing problem (CRP). More specifically, this class includes algorithms "equivalent" to the following algorithms (i-iii) developed by Mehlhorn, Preparata and Sarrafzadeh [7].

(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 multi-terminal net CRP with at most $2d_{max} - 1$ tracks.

The planar layouts generated by these algorithms and by their "equivalent" algorithms are three-layer wirable by the layer assignment algorithm given in [8]. Our approach is different. We make simple modifications to these layout algorithms and incorporate a simple wire assignment strategy to generate three-layer wirings under the knock-knee model. Consequently, we obtain simpler and faster algorithms that generate three-layer wirings with layouts similar to the ones generated by algorithms (i) - (iii). 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).

1. Introduction

The channel routing problem (CRP) has been recognized as one of the most important problems in VLSI design automation. The CRP problem is defined over the rectangular grid formed by lines $\{x = i \mid i \in Z\}$ and $\{y = j \mid 0 \le j \le h+1\}$. The horizontal grid lines $y = 0$ and $y = h+1$ are called the *boundaries* of the channel, and the horizontal grid lines $y = j$, $1 \le j \le h$, 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, $N = \{N_1, N_2, ..., N_m\}$, 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 [8] 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 $W = \{W_1, W_2, ..., W_m\}$ of the channel grid, such that each subgraph W_i connects all the terminals in net N_i. This definition implies that at each

‡ This research was supported in part by the National Science Foundation under Grant DCR - 8503163.
‡‡ Dr. Si-Qing Zheng new address is: Department of Computer Science, Louisiana State University, Baton Rouge, Louisiana, 70803-4020.

grid point in a planar layout there can be at most two wires, and 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 the knock-knee model, when two different wires share a grid point they either *cross* or form a *knock-knee*. The two types of knock-knees are given in figure 1.1. The horizontal (vertical) portions of a knock-knee are called the *horizontal (vertical) arms of the knock-knee.*

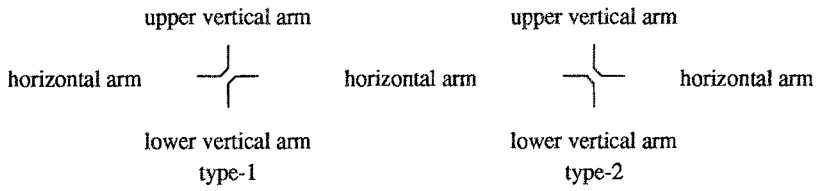

Figure 1.1: Knock-knees.

There are $k \geq 2$ conducting layers $L_1, L_2, ..., L_k$ available and L_{i+1} is stacked on top of L_i for $1 \leq i < k$. A *wiring* of a given layout $W = \{W_1, W_2, ..., W_m\}$ is a mapping that associates each edge of W_i, $1 \leq i \leq n$, to a layer such 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 $u > \mathbf{max}\{s, t\}$ or $u < \mathbf{min}\{s, t\}$. Contact cuts (called *vias*) can be established only at grid points. Vias allow a wire to change from one layer to the another. The 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.

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 $d_{max} = \mathbf{max}\{d(c)\}$, where d(c), the *local density* for the vertical cut (c,c+1), is the number of nets in N whose leftmost terminal is located to the left of vertical cut (c,c+1) and the rightmost terminal is located to the right of vertical cut (c,c+1). Clearly, the channel density d_{max} is a lower bound for the channel width in an optimal wiring for the CRP problem.

Preparata and Lipski [8] developed an efficient algorithm to generate a three-layer optimal wiring for the two-terminal net CRP. Their algorithm consists of two phases. In the first phase a minimum-track planar layout that satisfies certain properties is constructed. In the second phase the planar layout is transformed into a three-layer wiring by a powerful transformation, legal partition of the diagonal diagram induced by the layout. Recently, Mehlhorn, Preparata and Sarrafzadeh [7] developed another algorithm to construct a planar layout for this problem. The algorithm is conceptually simpler and the planar layout can also be three-layer wired by the algorithm in [8]. Another algorithm that finds a planar layout with d_{max} tracks for this problem appears in [10]; however, it is not known whether or not the layouts generated by this algorithm are three-layer wirable.

For the case when each net consists of at most three terminals, the algorithms in [7] and [9] generate a planar layout with no more than $\lfloor 3d_{max}/2 \rfloor$ tracks. For multi-terminal net channel routing problems, the algorithms in [4], [7] and [11] generate planar layouts with at most $2d_{max}$-1 tracks. The layouts generated by these algorithms are three-layer wirable by the algorithm given in [8]. The bounds $\lfloor 3d_{max}/2 \rfloor$ and $2d_{max}$-1 are believed to be best possible when one restricts to three layer wirable planar layouts. Recently, Gao and Kaufmann [2] showed that every

multiterminal net CRP has a planar layout with $3d_{max}/2 + O(\sqrt{d_{max}\log d_{max}})$ tracks. However, it is not known whether these layouts are three layer wirable or not.

Lipski [5] showed that there are planar layouts that are not three-layer wirable. Gonzalez and Zheng [3] 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([5]). Brady and Brown ([1]) showed that every planar layout is four-layer wirable by finding a legal partition (that satisfy some additional properties) of the diagonal diagram induced by the layout.

In this paper 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 currently best algorithms for the three layer channel routing problem. More specifically, this class includes algorithms "equivalent" to the following algorithms developed by Mehlhorn, Preparata and Sarrafzadeh [7].

(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 multi-terminal net CRP with at most $2d_{max}$ - 1 tracks.

The planar layouts generated by these algorithms and by their "equivalent" algorithms are three-layer wirable by the layer assignment algorithm given in [8]. Our approach is different. We make simple modifications to these layout algorithms and incorporate a simple wire assignment strategy to generate three-layer wirings. Consequently, we obtain simpler and faster algorithms that generate three-layer wirings with layouts similar to the ones generated by algorithms (i) - (iii). 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). As mentioned before, algorithms (i) - (iii) are currently the best algorithms for the three layer CRP problem.

2. Three-Layer Wiring Algorithms

Before we define the class of layouts algorithms we are interested in and our layer assignment strategy, we need to introduce additional notation. Let A be a layout algorithm that generates a layout by a single left-to-right column-by-column sweeping (scanning) of the 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 boundaries. When algorithm A process 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)*. There are no input wires for W(1) and for c > 1 *the input wires of W(c)* are the output wires for W(c-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 k 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, ..., v_m) is an *input (output) strand vector for W(c)* if wire W_i is a v_i-stranded input (output) wire in W(c). We use osv(W(c)) (isv(W(c)) to denote the output (input) strand vector for 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 final layout $W(c)$, i.e., once $W(c)$ is generated, the layouts $W(1), W(2), ..., W(c)$ 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 knock-knee of type-1 is below the knock-knee of type-2. 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 this additional property. One can easily develop algorithms "equivalent" to algorithms (i) - (iii) that satisfy properties (1) - (3). By "equivalent" we mean that the new algorithm has the same (asymptotic) worst case time complexity, and it never generates a layout with a number of tracks that exceeds the bounds stated in (i) - (iii). The planar layout algorithms given in [8] and [10] do not satisfy some of these properties and it seems that these algorithms do not have "equivalent" algorithms that satisfy properties (1) - (3).

Let A be any algorithm that satisfies properties (1) - (3). In what follows we define our algorithm, A^*, to construct a layout (similar to the one constructed by A) and find 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 $W(1), W(2), ...$ ($W^*(1), W^*(2), ...$) be the layout constructed by algorithm A (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 $k_1 \leq k_2$, 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. 2.1(a) */:
the interval [0,h+1] is labeled R_N;
:there is exactly one knock-knee in $W'(c)$ and its type is type-1 /* fig. 2.1(b) */:
let k be the track where the knock-knee is located;
the interval [0,h+1] is partitioned and labeled as follows: [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 its type is type-2 /* fig. 2.1(c) */:
let k be the track where the knock-knee is located;
the interval [0,h+1] is partitioned and labeled as follows: [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 $W'(c)$ /* fig. 2.1(d) */:
/* From property (3) we know that the knock-knees are of different types and that the knock-knee of type-1 is below the one of type-2 */
let k_1 (k_2) be the track where the knock-knee type-1 (type-2) is located;
/* From property (3) we know that $k_1 < k_2$ */
the interval [0,h+1] is partitioned and labeled as follows: [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

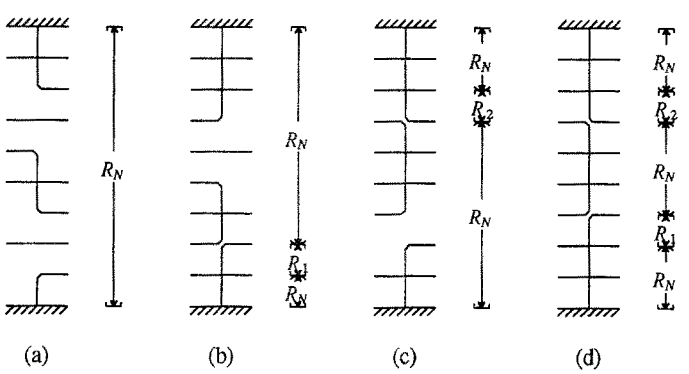

Figure 2.1: Labeling examples.

In the following figures the region inside the 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 its layer assignment are generated by algorithm A^* from $W'(c)$. Algorithm A^*

generates $W^*(c)$ by constructing sublayouts $W^*(c,I)$ for each interval I in $W'(c)$ uniformly labeled (i.e., with the same label). Let us now consider any uniformly labeled interval I. Initially we set the input wires in $W^*(c,I)$ to be the same as the output wires in $W^*(c\text{-}1,I)$ 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 $W^*(c\text{-}1,I)$. Depending on the label assigned to an interval, I, the remaining part of the layout $W^*(c,I)$ and the layer assignment for it 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 in this case is defined as follows. (Remember that the input wires in $W^*(c,I)$ are assigned to the same layers as the output wires in $W^*(c\text{-}1,I)$). 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 be assigned to the bottom layer).

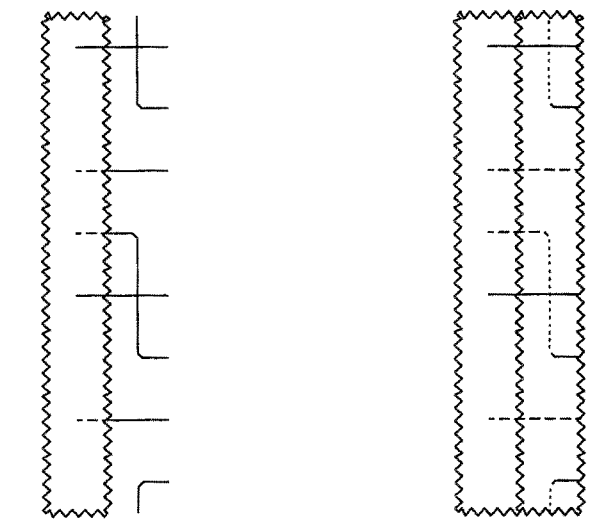

Figure 2.2: Example for case 1.

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 only one knock-knee in $W'(c,I)$, the type of knock-knee is type-1, and the knock-knee 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.

Algorithm A^* generates the layout $W^*(c,I) = W'(c,I)$. The layer assignment in this case 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.

Figure 2.3: Example for case 2.1.

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 in this case 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.

Figure 2.4: Example for case 2.2.1.

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 $W^*(c,I) \neq W'(c,I)$. $W^*(c,I)$ is $W'(c,I)$ after performing the transformation shown in figure 2.5.

Figure 2.5: Layout transformation for case 2.2.2.

The layer assignment in this case 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.

Figure 2.6: Example for case 2.2.2.

case 3: Interval I is labeled R_2.
Since case 3 is similar to case 2, it will be omitted.

end of procedure LAYOUT AND LAYER ASSIGNMENT

In figure 2.11 we give a planar layout constructed by algorithm (i) in [7]. The corresponding layout and the wiring constructed by our procedure is given in figure 2.12.

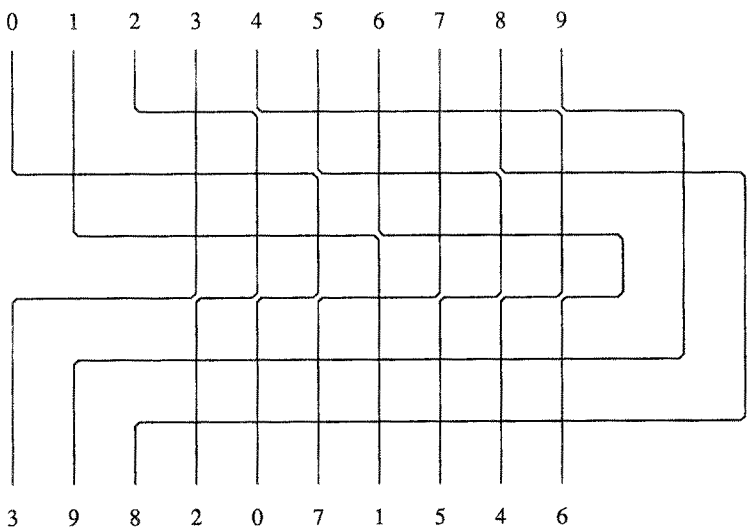

Figure 2.11: Layout constructed by algorithm (i) in [7].

Theorem 2.1: Let N, A, and A^* be as defined above. Algorithm A^* constructs a planar layout and its three layer wiring for N. 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 none of the layouts for W(1), W(2), ..., W(c) is modified. Algorithm A^* mimics this process. If the input strand vectors of W^*(c) and W(c) are identical, we know by property (2) and our

construction rules that the output strand vectors for $W^*(c)$ and $W'(c)$ are also identical. Therefore, it follows inductively (after a trivial proof for the base) that $W^*(c)$ and $W(c)$ have identical output strand vectors for every c. One can easily verify that W^* is a planar layout for N and that the three-layer wiring is valid. Therefore, Algorithm A^* constructs a planar layout and its three layer wiring for N; 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. □

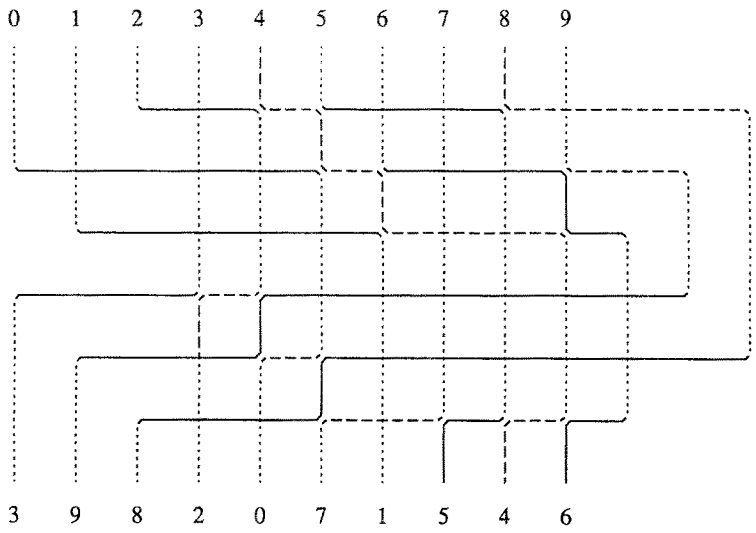

Figure 2.12: Layout and wiring constructed by our procedure.

3. Discussion

There are two major approaches to solve 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 three-layer assignment algorithm for planar layouts generated by conservative layout algorithms. This class of algorithms includes simple variations of well known algorithms for the channel routing problem. Our approach consists of making simple modifications to the layout algorithm and incorporating a simple wire assignment strategy to generate three-layer wirings. Consequently, we obtain simpler and faster algorithms that generate three-layer wirings for layouts similar to the ones generated by algorithms (i) - (iii). Our algorithms are faster and conceptually simpler because there is no need to construct diagonal diagrams and legal partitions. The channel width of the wirings generated is identical to that of the planar layouts (i) - (iii). Algorithms (i) - (iii) are currently the best algorithms for the three layer channel routing problem.

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 have a simple equivalence 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, wiring 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 [7] can be wired in three layers by using the layer assignment algorithm given in [8]. 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. We should point out that the time complexity for our algorithm is identical to that of the procedures given in [7]. Note that one does not need to output at each step unit wire segments in each of the tracks.

Bibliography

[1] Brady, M. L. and D. J. Brown, "VLSI Routing: Four Layers Suffice", Advances in Computing Research, vol. 2, 1984.

[2] Gao, S. and M. Kaufmann, "Channel Routing of Multiterminal Nets", Proceedings of the 28th Symposium on Foundations of Computer Science, pp 316-325, 1987.

[3] Gonzalez, T. and Zheng, S.-Q., "Wirability of Planar Layouts", Technical Report, # 87-11, CS Dept., UC Santa Barbara, Aug. 1987.

[4] Gonzalez, T. and Zheng, S.-Q., "Three-Layer Channel Routing of Multi-terminal Nets", Technical Report, # 87-13, CS Dept., UC Santa Barbara, Aug. 1987.

[5] Lipski, W. Jr, "An NP-complete Problem Related to Three-layer Channel Routing", Advances in Computing Research, vol. 2, 1984.

[6] Mehlhorn, K. and F. P. Preparata, "Routing Through a Rectangle", *J. ACM*, vol. 33, no. 1, 1986.

[7] Mehlhorn, K., F. P. Preparata and M. Sarrafzadeh, "Channel Routing in Knock-Knee Mode: Simplified Algorithms and Proofs", *Algorithmica*, no. 1, 1986.

[8] Preparata, F. P and W. Lipski, Jr, "Optimal Three-layer Channel Routing", *IEEE Transaction on Computer.*, vol .33, no. 5, 1984.

[9] Preparata, F. P. and M. Sarrafzadeh, "Channel Routing of Nets Bounded Degree", *VLSI: Algorithms and Architectures*, North-Holland, 1984.

[10] Rivest, R. L., A. Baratz and G. Miller, "Provably Good Channel Routing Algorithms," in Proc. 1981 Carnegie-Mellon Conf on VLSI, Oct. 1981, pp. 153-159.

[11] Sarrafzadeh, M. and F. P. Preparata, F. P., "Compact Channel Routing of Multiterminal Nets", *Annals of Discrete Mathematics*, no. 25, April 1985, pp. 255-279.