VLSI Design
1994, Vol. 1, No. 3, pp. 233–242
Reprints available directly from the publisher
Photocopying permitted by license only

An Optimum Channel Routing Algorithm in the Knock-knee Diagonal Model*

XIAOYU SONG
Dipartimento di Informatica, Università di Pisa, Pisa, Italy

(Received July 30, 1990, Revised August 11, 1992)

Channel routing problem is an important, time consuming and difficult problem in VLSI layout design. In this paper, we consider the two-terminal channel routing problem in a new routing model, called knock-knee diagonal model, where the grid consists of right and left tracks displayed at $+45°$ and $-45°$. An optimum algorithm is presented, which obtains $d + 1$ as an upper bound to the channel width, where d is the channel density.

Key Words: *VLSI layout, Channel Routing, Diagonal Model, Algorithms*

INTRODUCTION

Routing is a crucial problem in the VLSI layout design automation process. The process of routing is generally divided into two stages: *global routing,* which is global assignment of the wiring paths for each net, and *detailed routing,* which is detailed wiring in individual routing regions. Detailed routing is an important, time consuming and difficult problem. The Channel Routing Problem (CRP) arises in the detailed routing. It consists of connecting *terminals* belonging to *nets,* which are displayed on two opposite sides (*entry* and *exit lines*) of a rectangular channel.

Channel routing problems have been extensively studied in different traditional routing models. In particular, we recall Manhattan Model (MM) [1], the Knock-knee model (KK) [2]. In the Manhattan model, two wires may share a grid point only by crossing at that point, but the wires are not allowed to overlap. In the knock-knee model wires may share a grid point either by crossing or by bending at that point, but two wires are not allowed to overlap.

Besides the usual square grid, other tessellations of the plane, such as the hexagonal, octosquare grid, etc., have been proposed for routing. In this paper, we consider a new routing model, called *Knock-Knee Diagonal model* (KKD), where the grid consists of right and left tracks displayed at $+45°$ and $-45°$. We present an optimum algorithm which obtains $d + 1$ as an upper bound to the channel width, where d is the channel density.

DEFINITIONS OF PROBLEMS

Let P and Q denote two sets of terminals on the entry and exit lines, respectively, $|P| = |Q| = m$. The terminals are identified by their integer coordinates along the lines. A two-terminal net $N = (p, q)$ is a pair of integers where $p \in P$ and $q \in Q$. A net is *trivial* if $p = q$. A two-terminal CRP is specified as a set $\{N_1, \ldots, N_n\}$ of n nets. The main goal of a CRP is to construct a layout in a channel of minimum width.

A fundamental parameter is the *density* d defined as follows. Let $c \in P \cup Q$ and $x = c + \varepsilon$ $(0 < \varepsilon < \frac{1}{4})$, and let $d(x)$ be the number of nets (p, q) which cross the vertical line x, i.e. $((p < x) \wedge (q > x))$ or $((p > x) \wedge (q < x))$. The density of a CRP is $d = \max\{d(x)\}$. In MM and KK the density d is a trivial lower bound to the channel width.

Instead of using a horizontal-vertical grid, we consider a diagonal grid for routing. The word "*diagonal*" means a slope of $\pm 45°$. Diagonal line segments which are displayed at $+45°$ and $-45°$ are called

*This work has been supported in part by Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo of CNR, Italy.

right and *left tracks*. The grid consists of diagonal line segments which are composed of the left tracks and the right tracks. A channel on the diagonal grid is shown in Figure 1. The entry terminals and exit terminals are located at unit distance on two horizontal (*entry* and *exit*) lines. The distance between these two lines, called *vertical width w*, is a multiple of the unit; hence terminals lie at the intersections of left and right tracks.

To judge the interest on the routing models on a diagonal grid, let us compare them with the traditional routing models. As a variation of MM, the routing model: Diagonal Model (DM), has been proposed in [3]. There have been some interesting results on CRPs in DM [3, 4, 5, 6, 7, 8]. Note that some problems which are very difficult in MM, such as the *shift-by-1* problem, where each entry terminal i must be connected to the exit terminal $i + 1$ (or $i - 1$), can be solved easily in DM. In MM it requires $\Omega(\sqrt{n})$ tracks although its density is one. In DM the problem can be optimally embedded in a channel of vertical width equal to one. Moreover, the whole class of problems, *shift-by-i,* $1 \leq i \leq n$, can be optimally solved in DM using a channel of vertical width equal to i. *Shift-by-0* is also optimally laid out with vertical width one.

In the theoretical investigation of two-terminal CRPs in MM, the best algorithm for MM [1, 9] obtains $w = d + O(f)$ where f is the *flux* and $f \leq \sqrt{n}$, n is the number of nets. In particular, the worst case of f usually holds for dense problems [10]. The best DM router [8] yields $w \leq 2d + 1$, for d even, or $w \leq 2d + 3$, for d odd, where the difficult parameter f (channel flux) disappears. We can only use the density d to measure the difficulty of problem.

Due to the difficulty of MM, other routing models have been proposed. Among them, the knock-knee model is the most important one due to its great advantage of the saving in the area. Recall that the two-terminal CRP, which is NP-complete in MM, can be solved optimally in KK, namely, $d + 1$ is an upper bound to w. There exists a complete theory for CRPs in KK [2, 11], while no result in the knock-knee model on a diagonal grid. In order to fill up this lacuna, we investigate in this paper the efficiency of the knock-knee diagonal model for CRPs. In KKD wires may share a grid point either by crossing or by bending at that point but cannot share segments. As in KK, it may require more than two layers to obtain a good layout. We shall show that there is a similar optimal result for the two-terminal CRP in KKD, namely, $d + 1$ as an upper bound to the channel width, where d is the channel density.

Note that in the study of routing problems, for the same terminal locations, the distance between two adjacent tracks is smaller by a factor of $\sqrt{2}$ in KKD than in the traditional routing models. If the same track distance is to be maintained for technological reasons, then the unit distance between terminals and the vertical size of the channel for a given routing must be enlarged by a factor of $\sqrt{2}$. A general remark on the diagonal model is that the conductive tracks of each layer are closer than those in the traditional models, by a factor $1/\sqrt{2}$. However, this may not seem to cause problems in the current and future technology.

As mentioned before, in the traditional routing models, such as MM and KK, the density of problem d is a trivial lower bound to the channel width w. In KKD an immediate lower bound to w is $\lceil d/2 \rceil$ [3]. It has been shown [8] that d is a non-trivial lower bound to w in DM and in KKD.

THE ROUTING ALGORITHM

We give an optimal algorithm which never requires a channel width more than $d + 1$. The whole algo-

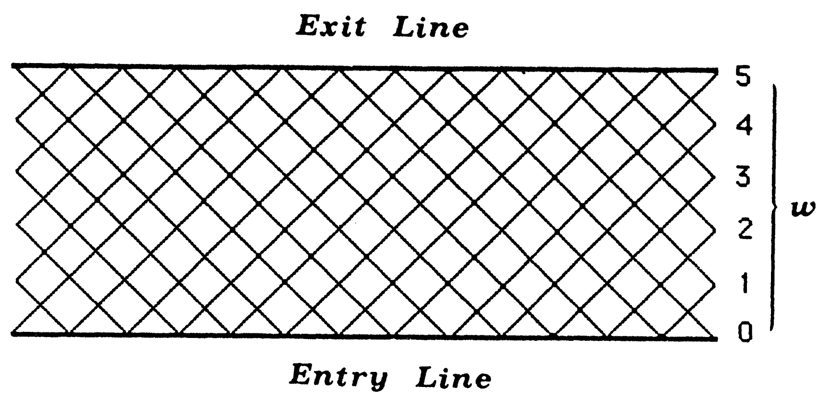

FIGURE 1 A channel on a diagonal grid.

rithm consists of three parts:

1. Construction of chains;
2. Routing the CRP in a channel of width $\lceil 3/2\, d \rceil$;
3. Compacting the obtained routing into a channel of width $d + 1$.

In what follows, we discuss these parts one by one, and finally give the complete routing algorithm.

Construction of Chains

We represent nets with arrows. For a net (p, q), if $p \neq q$, its arrow is a directed horizontal line segment from the starting point with abscissa p to the ending point with abscissa q; and if $p = q$, its arrow is a vertical arrow with one unit length. Let Π be an instance of two-terminal CRP.

Definition 1. A *chain* is a sequence of nets $C = (p_1, q_1), (p_2, q_2), \ldots, (p_n, q_n)$ such that any two nets do not overlap with each other, i.e. $p_{i+1} \geq \max(p_i, q_i)$ and $q_{i+1} \geq \max(p_i, q_i)$, and $p_i \neq q_i$, $i = 1, \ldots, n$.

The main idea of our routing process consists of the following steps:

(i) Partitioning the arrow set into chains $E_1, E_2, \ldots, E_d$.
(ii) Routing each pair of chains E_i and E_{i+1} (i odd) successively in a stripe of width 3.

Definition 2. For each chain E_i, we define a set T_i of trivial nets, which includes the original trivial nets of Π, and a trivial net (s, s) for each net $S = (r, s) \in E_j$, $j < i$; and for each net $S = (s, r) \in E_j$, $j > i + 1$.

In the process of routing there exist some nets previously routed and occupying their final positions, and they may block the routing of some other nets not yet handled (see Figure 2). Therefore, some nets have to be prolonged or contracted for the time being in order to have other wires pass. More specifically, consider two nets $N_1 = (p, q)$ and $N_2 = (q, r)$ such that $N_1 \in E_i$ and $N_2 \in E_j$. If $i < j$ and $p < q$, after the routing of N_1, N_1 becomes a trivial net (Definition 2) and must occupy the position of q, this causes a conflict (see Figure 2), because N_2 has not been routed and is also a trivial net. To avoid such conflicts, we introduce the concept of *target point.* For N_1, we replace N_1 by two nets $N_1' = (p, t)$ and $N_1'' = (t, q)$, where t is a proper target point defined below, $t > q$. N_1' and N_1'' are respectively called the *extended* and *compensation nets* of N_1.

Definition 3. The *target point* t of a net (p, q) with $p < q$ is the minimum integer $t \geq q$ such that the set of nets still to be inserted in the chains does not contain either a net (t, r) with $r \leq t$ or a net (r, t) with $r < t$; and the set of nets already inserted in the chains does not contain a net (r, t) (see Figure 2).

Using the extended and compensation nets, we can avoid the conflicts between the nets handled previously and successively. The compensation net is used to return to the original destination. Note that after the construction of the extended and compensation nets, it may well be that there are two arriving and two leaving nets at point t, but this does not cause

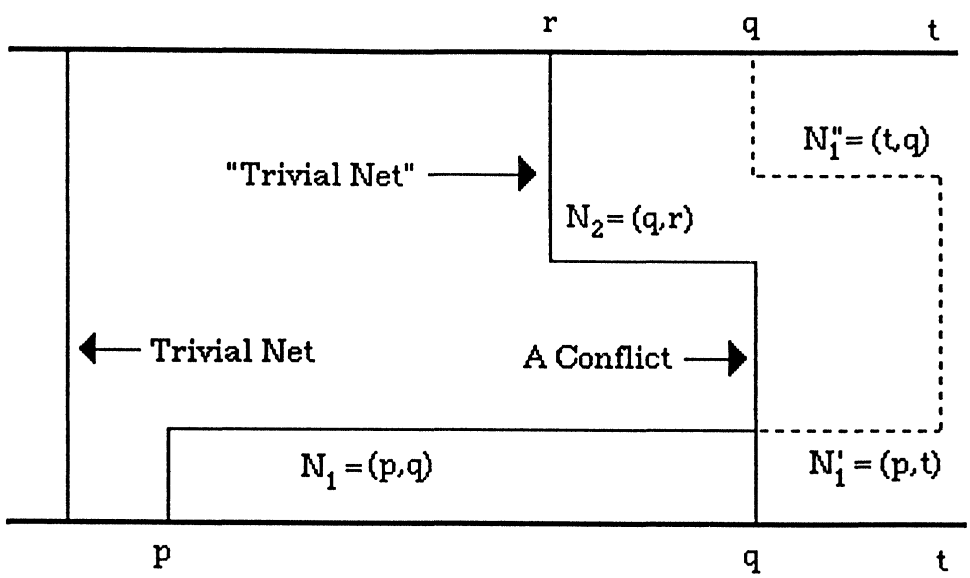

FIGURE 2 Illustration of the definitions of trivial, extended, and compensation nets.

the inconsistency because they are routed in different routing phases.

We now construct the chains of a CRP Π. Let Π = {N_1, N_2, . . . , N_n} be a CRP and E_1, E_2, . . . , E_d be the chains.

The Chain Construction Algorithm

```
Function Target(t);
if (¬∃(t, r) ∈ Π with r ≤ t) ∧ (¬∃(r, t) ∈ Π with
r < t) ∧ (¬∃(r, t) ∈ Ek with k < i)
then Target := t
else Target(t + 1);

begin {Algorithm}
i := 0;
while (Π ≠ Ø) ∧ (∃N ∈ Π is not trivial) do
begin i := i + 1; j := 1;
  while j < n do
  if (∃(j, q) ∈ Π with q > j) ∧ (¬∃(r, j) ∈ Π
  with r < j)
    then while ∃(j, q) ∈ Π with q > j do
         (j, Target(q)) → Ei;
         if Target(q) ≠ q
         then begin
              Change(j, q) to (j, Target(q))
              in Π; (Target(q), q) → Π
              end;
         j := Target(q)
         end {while}
    else if ∃(p, j) ∈ Π with p > j) ∧ (¬∃(j, r) ∈
       Π with r < j)
      then while (∃(p, j) ∈ Π with p > j) do
           (p, j) → Ei; j := p
           end; {while}
    j := j + 1
  end; {while}
  Π := Π − Ei
  end {while}
end. {Algorithm I}                                □
```

The chains are built from left to right packing the nets as tightly as possible. Any pair of nets with adjacent span [*a*, *b*], [*b*, *c*] will be inserted in the same chain. For instance, using the chain construction algorithm, we can attain a set of chains for a CRP Π = {⟨1,5⟩, ⟨2,6⟩, ⟨3,8⟩, ⟨5,9⟩, ⟨6,3⟩, ⟨7,2⟩, ⟨8,15⟩, ⟨10,10⟩, ⟨11,14⟩, ⟨12,19⟩, ⟨13,11⟩, ⟨14,12⟩, ⟨16,16⟩, ⟨17,20⟩, ⟨18,21⟩, ⟨19,13⟩}. The extended, compensation nets and their chains are indicated with dashed and dotted arrows, respectively, in Figure 3. For example, the extended and compensation nets of net ⟨2,6⟩ are nets ⟨2,11⟩ and ⟨11,6⟩, respectively. Note that the target point t = 11 because of the existence of nets ⟨**7**,2⟩, ⟨3,**8**⟩, ⟨5,**9**⟩ and ⟨**10**,**10**⟩.

The following proposition guarantees that the extended and compensation nets do not increase the density of the problem.

Proposition 1 [6]. The number of chains in the extended construction of a CRP Π is equal to *d*.

Routing the CRP in a Channel of Width [3/2 *d*]

The channel consists of stripes S1, . . . , $S_{[d/2]}$, each of width 3. The basic connections in a stripe of channel are two busses, called *lower bus* and *upper bus*, and *0-connections* (for "trivial" nets). Each bus goes through the channel. The busses introduce a division of the points on the stripe borders into two groups, called *1-points* and *2-points*, respectively (see Figure 4). The 0-connection of point *p* (connection of *p* for short) is defined as shown in Figure 4, depending on the position of *p*.

The chains E_1, . . . , E_d are grouped in pairs (E_i, E_{i+1}) (i is odd), and each pair (E_i, E_{i+1}) is routed in stripe $S_{[i/2]}$. In general, the "trivial nets" of E_i and E_{i+1} use 1- or 2-connections, while the long connections of E_i and E_{i+1} are routed by using the lower and upper bus, respectively.

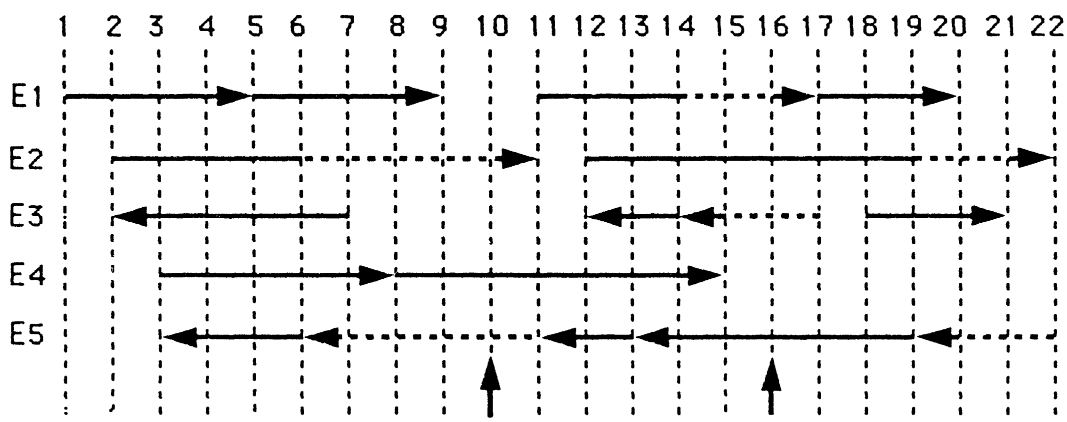

FIGURE 3 The construction of chains for a CRP.

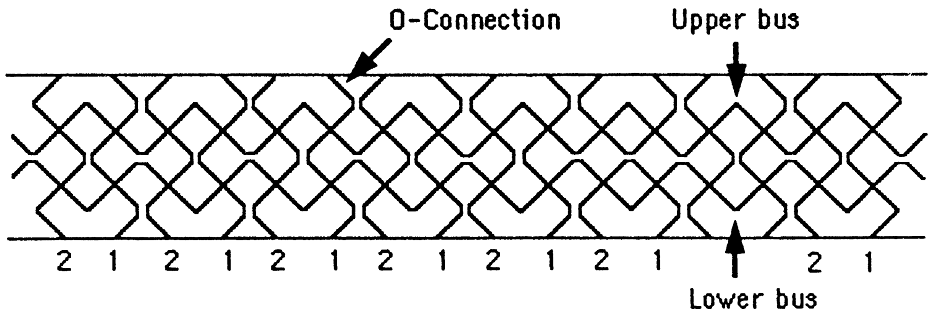

FIGURE 4 Basic connections in a stripe.

For an odd i, $1 \le i \le d$, let

$R^{low} = \{(p, q): (p < q) \wedge ((p, q) \in E_i)\}$,
$R^{up} = \{(p, q): (p < q) \wedge ((p, q) \in E_{i+1})\}$,
$L^{low} = \{(p, q): (p > q) \wedge ((p, q) \in E_i)\}$,
$L^{up} = \{(p, q): (p > q) \wedge ((p, q) \in E_{i+1})\}$.

Namely, R^{low} and L^{low} contain the right and left nets of E_i, respectively; R^{up} and L^{up} contain the right and left nets of E_{i+1}, respectively.

We now construct the routing using the five routing rules presented below. We enumerate all the possible cases in the routing of E_i and E_{i+1} in stripe $S_{\lceil i/2 \rceil}$.

Rules 1 to 4 deal with the cases in which if an extended net is in E_i, its compensation net will be in E_j, $j > i + 1$, while the case in which such a net is in E_{i+1} is treated in rule 5.

Rule 1. Routing $(p, q) \in R^{low}$

1. Connect the entry point p of the current stripe to the lower bus. There are three cases:

(a) p is a 2-point.
From p start with the left track and move on the 0-connection of p to meet the lower bus (p′ in Figure 5.1).

(b) (p is a 1-point) $\wedge$ $(\neg\exists(r, p) \in R^{low})$.
From p start with the right track and move on the 0-connection of p to meet the lower bus (p″ in Figure 5.1).

(c) (p is a 1-point) $\wedge$ $(\exists(r, p) \in R^{low})$ (i.e. nets (r, p) and (p, q) are adjacent in R^{low}).
This case is covered in the following point (3), while treating the connection of the end point of (r, p).

2. Proceed on the lower bus to the right.

3. Connect the lower bus to the exit point q of the current stripe. We have four cases:

(a) q is a 2-point.
Use the routing shown in Figure 5.1.

(b) (q is a 1-point) $\wedge$ $(\neg\exists(q, r) \in R^{low})$ $\wedge$ $(\neg\exists(r, q + 1) \in L^{low})$.
Use the routing shown in Figure 5.2.

(c) (q is a 1-point) $\wedge$ $(\exists B = (q, r) \in R^{low})$.
Let A = (p, q). The routing is shown in Figure 5.3.

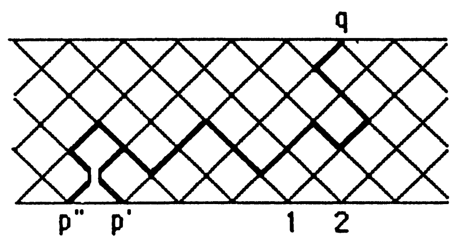

FIGURE 5.1 Rule 1. Case 1(a), 1(b), and 3(a).

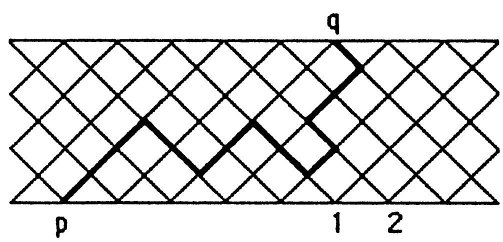

FIGURE 5.2 Rule 1. Case 3(b).

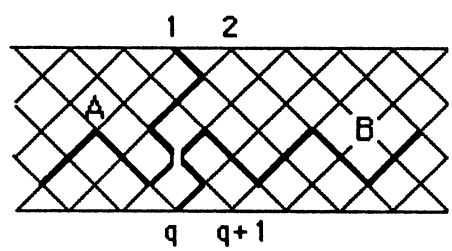

FIGURE 5.3 Rule 1. Case 3(c).

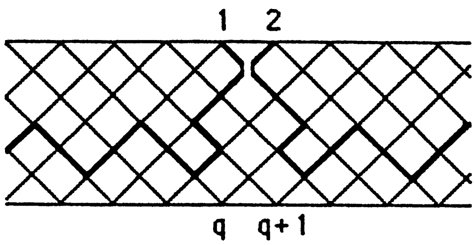

FIGURE 5.4 Rule 1. Case 3(d).

(d) (q is a 1-point) $\wedge$ ($\exists$B = (r, q + 1) $\in$ L^{low}). Let A = (p, q). The routing is shown in Figure 5.4.

Rule 2. Routing (r, s) $\in$ L^{low}

This rule is symmetrical to Rule 1, exchanging left and right, and 1-points with 2-points.

Rule 3. Routing (p, q) $\in$ R^{up}

1. Connect the entry point p of the current stripe to the upper bus. There are four cases:

(a) p is a 1-point.
From p start with the right track and move on the 0-connection of p to meet the upper bus (p′ in Figure 5.5).

(b) (p is a 2-point) $\wedge$ ($\neg\exists$(r, p) $\in$ R^{up}).
From p start with the left track and move on the 0-connection of p to meet the upper bus (p″ in Figure 5.5).

(c) (p is a 2-point) $\wedge$ ($\exists$(r, p) $\in$ R^{up}) $\wedge$ ($\neg\exists$(q − 1, r) $\in$ L^{up}).
This case is covered in the following point (3), while treating the connection of the end point of (r, p).

(d) (p is a 2-point) $\wedge$ ($\exists$B = (p − 1, r) $\in$ L^{up}). Let A = (p, q). The routing is shown in Figure 5.6.

2. Proceed on the upper bus to the right.
3. Connect the upper bus to the exit point q of the current stripe. We have three cases:

(a) q is a 1-point.
Use the connections of Figure 5.5.

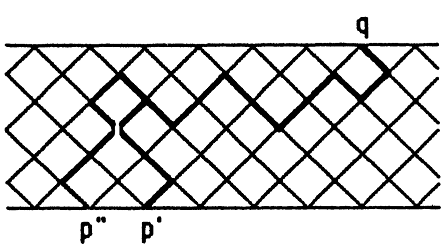

FIGURE 5.5 Rule 3. Case 1(a), 1(b), and 3(a).

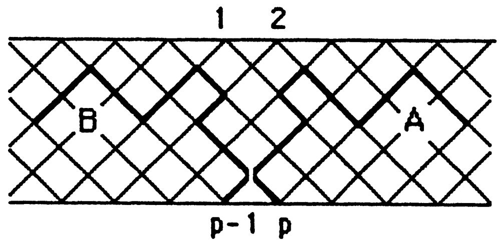

FIGURE 5.6 Rule 3. Case 1(d).

(b) (q is a 2-point) $\wedge$ ($\neg\exists$(q, r) $\in$ R^{up}).
Use the routing shown in Figure 5.7.

(c) (q is a 2-point) $\wedge$ ($\exists$B = (q, r) $\in$ R^{up}).
Let A = (p, q). The routing of A and B are shown in Figure 5.8.

Rule 4. Routing (r, s) $\in$ L^{up}

This rule is symmetrical to Rule 3, exchanging left with right, and 1-points with 2-points.

Rule 5. Routing an extended net in R^{low} and its compensation net in L^{up} i.e. (N1 = (p, q) $\in$ R^{low}) $\wedge$ (N2 = (q, r) $\in$ R^{low}) $\wedge$ (N1′ = (q, v) $\in$ L^{up}) $\wedge$ (N3 = (s, q) $\in$ L^{up}) $\wedge$ (N1′ is the compensation of N1). N2 and/or N3 may not be present. There are two cases:

(a) q is a 1-point.

Use the global routing for N1 and N1′, and the routing for N2 and N3 shown in Figure 5.9. Note that the global routing for N1 and N1′ uses a portion of the 0-connection of q. If there are not N2 and N3, the 0-connection of q is modified by using a portion of two buses as shown Figure 5.10.

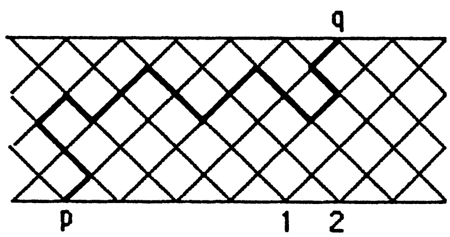

FIGURE 5.7 Rule 3. Case 3(b).

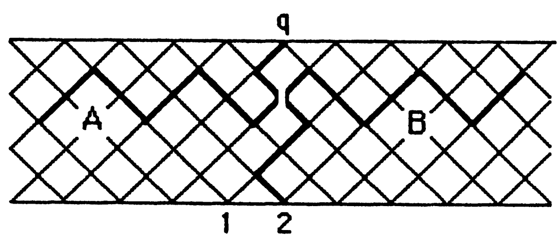

FIGURE 5.8 Rule 3. Case 3(c).

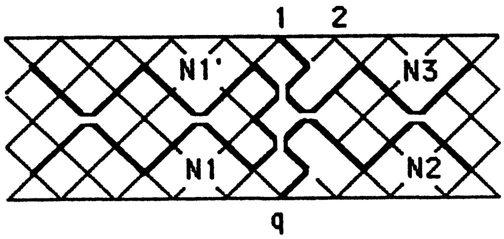

FIGURE 5.9 Rule 5. Case (a).

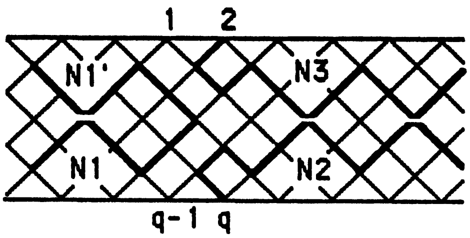

FIGURE 5.10 The modification of 0-connection of q in Rule 5.

(b) q is a 2-point.

Use the global routing for N1 and N1′, and the routing for N2 and N3 shown in Figure 5.11. This does not cause any conflicts.

A layout of an arbitrary two-terminal CRP can be obtained as the layout of E1, E2, . . . , Ed built one on top of the other starting at the entry line, therefore completing the whole routing. From the above discussion, we have:

Proposition 2. Two-terminal CRP can be solved in a channel of width $w = [3/2\ d]$.

Proof. The whole CRP is transformed into a set of subproblems. Each subproblem is the problem of routing a pair of chains. We only need to enumerate all the cases for a pair of chains (E_i, E_{i+1}), *i* is odd, routed in the same stripe of channel. All the different cases for *one* net are covered by rules 1 to 4. Those for *two* nets are covered by rules 1 to 5. This implies the completeness of the algorithm.

Based on the properties of rules 1 to 5, the correctness of the algorithm is proved by noting that each net N = (p, q) can only use the 0-connections of p and q, and/or the lower or upper bus in the stripe reserved for N. There is no conflict among the routing of nets. The channel width is attained by recalling that each two consecutive chains are routed in a stripe of width 3. □

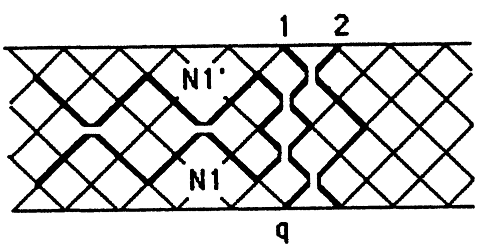

FIGURE 5.11 Rule 5. Case (b).

Compaction of the Routing

We now show how to compact the obtained routing to have a channel of width $d + 1$. Let H be the obtained routing which consists of stripes $S_1, \ldots, S_{[d/2]}$, each of width 3.

The Compaction Algorithm

Input: Routing H;
Output: Routing R;
Step 1: Eliminate the upper stripe of width ½ from S1;
Step 2: **for** i := 2 **to** $\lceil d/2 \rceil - 1$ **do**
Eliminate the upper and the lower stripes of width ½ from each stripe Si;
Step 3: Eliminate the lower stripe of width ½ from $S_{[d/2]}$. □

In order to explain the compacting process more explicitly, we introduce two auxiliary horizontal lines, called L_i^l and L_i^u, in each stripe Si, $1 \le i \le [d/2]$. These lines pass through the lowest points of the lower bus and the highest points of the upper bus, respectively. Note that the auxiliary horizontal lines do not belong to the routing. In addition, for each two stripes S_i, S_{i+1}, we define the following sets of points for each L_i^l and L_i^u, $1 \le i \le [d/2]$.

(i) $A_{i+1}^l = \{a_1, a_2, \ldots, a_n\},$

$A_i^u = \{a_1', a_2', \ldots, a_n'\},$

such that a_i and $a_{i'}$ $(1 \le i \le n)$ are respectively located on the lines L_{i+1}^l and L_i^u, and do not belong to the lower bus of S_{i+1}, and to the upper of S_i, respectively. The points in each pair $(a_i, a_{i'})$ have the same abscissa.

(ii) $B_{i+1}^l = \{b_1, b_2, \ldots, b_n\},$

$B_i^u = \{b_1', b_2', \ldots, b_n'\},$

such that b_i and b_i' $(1 \le i \le n)$ are respectively located on the lines L_{i+1}^l and L_i^u, and belong to the lower bus of S_{i+1} and to the upper of S_i, respectively. The points

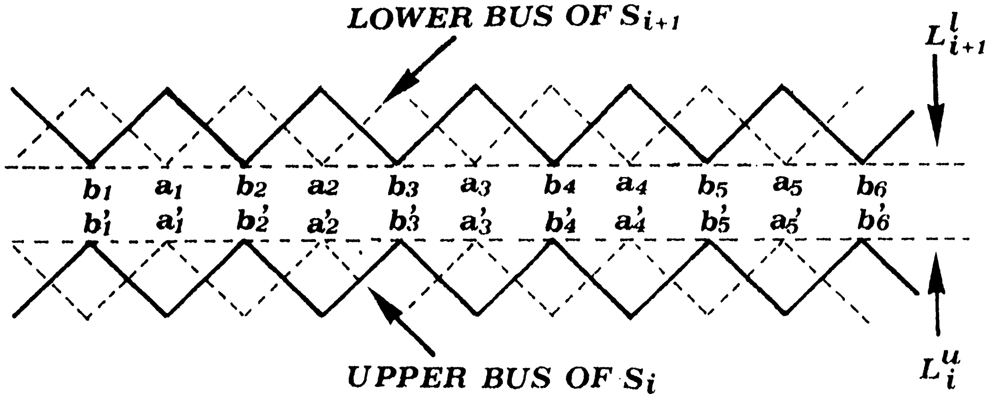

FIGURE 6.1 Definitions of two adjacent L^u_i and L^l_{i+1}.

in each pair (b_i, b_i') have the same abscissa (see Figure 6.1).

Considering two adjacent stripes S_i, S_{i+1}, $1 \le i \le [d/2]$, we have the following compaction transformation rules.

Rule 1. Compaction of B^l_{i+1} and B^u_i.

Put each two points (b_i, b_i'), $1 \le i \le n$, together by sharing one point with a knock-knee shown in Figure 6.2.

Rule 2. Compaction of A^l_{i+1} and A^u_i.

Put each two points (a_i, a_i'), $1 \le i \le n$, together by sharing one point according to the different cases shown in Figure 6.3. Note that these cases occur when there are connections passing through a_i and a_i'. If this does not occur the compaction is done simply by putting the corresponding points a_i and a_i' together.

The correctness of the compaction process can be easily proved by noting that there are no conflicts among the compacted points.

From the above discussion, we can see that, after the compaction of H, we obtain a new routing R in which the channel consists of stripes $S_2, \ldots, S_{\lceil d/2 \rceil - 1}$, of width 2, the stripes S_1 and $S_{\lceil d/2 \rceil}$, of width 5/2.

The Complete Routing Algorithm

We now give the complete routing according to the above discussions.

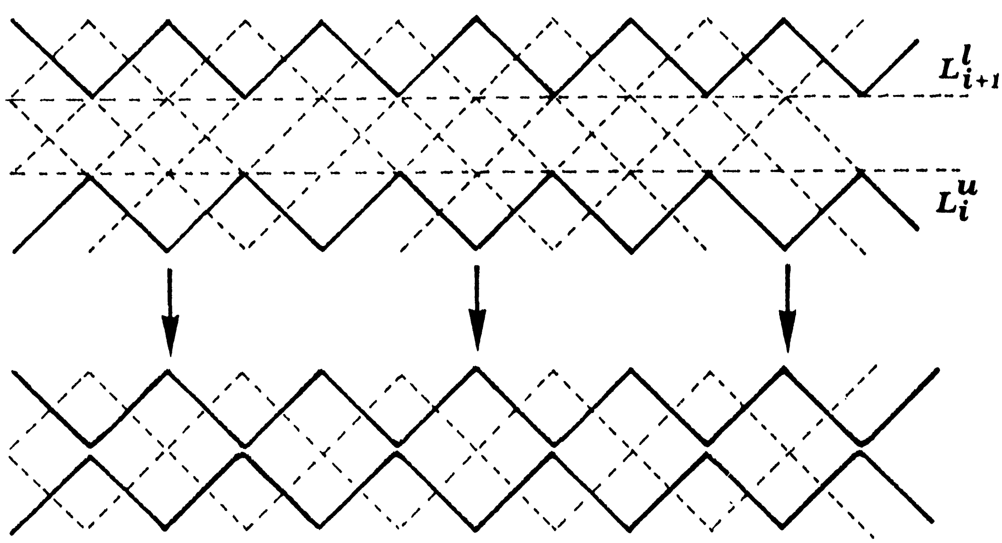

FIGURE 6.2 Illustration of Rule 1.

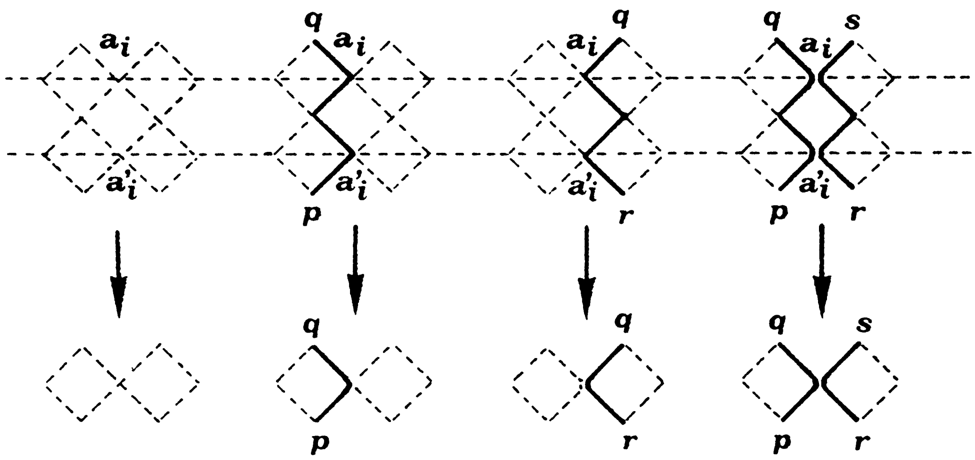

FIGURE 6.3 Different cases of Rule 2.

The Optimal Channel Routing Algorithm

Input: A CRP $\Pi = \{N1, N2, \ldots, Nn\}$;
Output: Routing R;
begin
baseline := 0; {Zero is the vertical coordinate of the entry line}
Use the chain construction algorithm to construct chains $E_1, E_2, \ldots, E_d$;
for i := 1 **to** $\lceil(d + 1)/2\rceil - 1$ **do**
begin
Use rules 1–5 to route the nets of a pair of chains (E_i, E_{i+1}) in the current stripe of height 3;
Use 0-connections to route nets in E_k, where $(k \neq i$ and $k \neq i + 1)$, and trivial nets to reach the exit line of the current stripe;
baseline := baseline + 3
end;
if d is odd
then
begin
Use rules 1–4 to route the nets of a chain E_d in the current stripe of height 2;
Use 0-connections to route nets in E_k $(k \neq d)$, and trivial nets to reach the exit line of the current stripe
end;
Use the compaction algorithm to compact the obtained routing
end. □

From the above discussions and proposition 2, we conclude with the following theorem:

Theorem 1. The presented algorithm produces a routing for two-terminal CRP with the channel width $d + 1$ in KKD, where d is the channel density. The running time of the algorithm is $O(md)$, where m is the length of channel, and d is the density.

Applying the algorithm to the example of Figure 3, we obtain the routing in a channel of width 6, where $d = 5$, as shown in Figure 7.

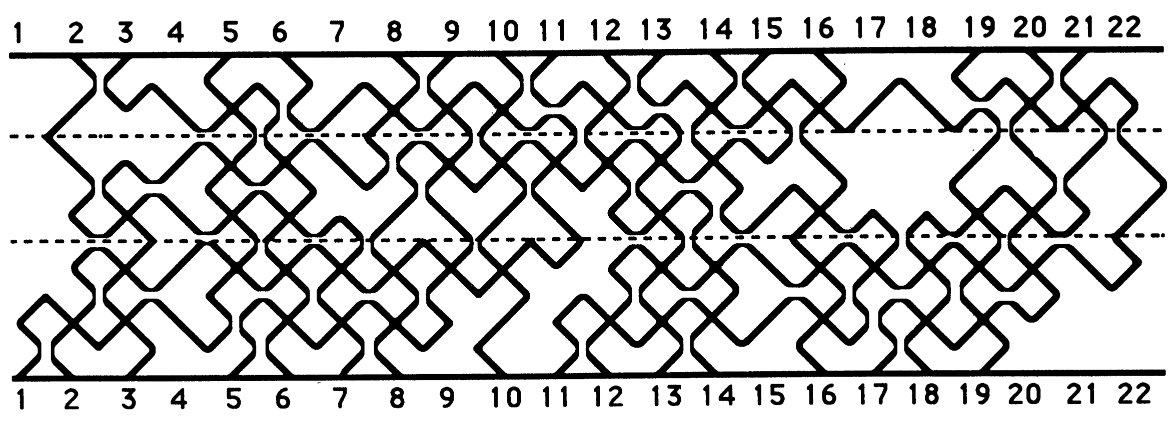

FIGURE 7 The routing of an example CRP.

CONCLUSIONS AND OPEN PROBLEMS

In this paper, we have studied the two-terminal channel routing problem in the knock-knee diagonal model. We presented an optimum routing algorithm which obtains $d + 1$ as an upper bound to the channel width, where d is the channel density.

There are still some open problems. The diagonal routing often requires a lot of vias. This may be a serious problem in practice. A comparative study of traditional and diagonal models in this respect is still under way. On the other hand, the only thing we know is that any layout can be wired in four layers [11]. To extend the above algorithm to solve the problem in two or three layers is still open.

Acknowledgments

The author is grateful to Professor Fabrizio Luccio for his supports and suggestions. Thanks are due to the unknown referees for their suggestions leading to essential improvements of this paper.

References

[1] B.S. Baker, S.N. Bhatt, and F.T. Leighton, "An approximation algorithm for Manhattan routing," Proc. 15th Ann. ACM Symp. on Theory of Computing, pp. 477–486, 1983.
[2] K. Mehlhorn, F.P. Preparata, and M. Sarrafzadeh, "Channel Routing in Knock-knee Mode: Simplified Algorithms and Proofs," *Algorithmica,* vol. 1, pp. 213–221, 1986.
[3] E. Lodi, F. Luccio and L. Pagli, "A Preliminary Study of a Diagonal Channel Routing Model," *Algorithmica,* vol. 4, pp. 585–597, 1989.
[4] E. Lodi, "Routing Multiterminal Nets in a Diagonal Model," Proc. Conf. Information Science and Systems, Princeton, pp. 899–902, 1988.
[5] E. Lodi, F. Luccio and L. Pagli, "A 4d Channel Router for A Two Layer Diagonal Model," Proc. A.W.O.C. 88, Corfu, Greece 1988. In LNCS 319, Springer Verlag, New York 1988.
[6] E. Lodi, F. Luccio and L. Pagli, "A Good Channel Router for A Diagonal Model," Technical Report, TR-14/89, Dipartimento di Informatica, University of Pisa, April 1989.
[7] E. Lodi, F. Luccio, and L. Pagli, "Channel Routing for Strictly Multiterminal Nets," *Integration, the VLSI Journal* Vol. 8, No. 2, 143–153, 1989.
[8] E. Lodi, F. Luccio and X.Y. Song, "A 2d Channel Router for the Diagonal Model," Technical Report, TR-38/89, Dipartimento di Informatica, University of Pisa, December 1989.
[9] F.T. Leighton, "A survey of problems and results for channel routing," Proc. A.W.O.C. 86, Loutraki, Greece, 1986.
[10] D.J. Brown, and R.L. Rivest, "New Lower Bounds on Channel Width," Proc. CMU Conference on VLSI Systems and Computations, Computer Science Press 1981, pp. 178–185.
[11] M. Brady, and D.J. Brown, "VLSI routing: four layers suffice," *Advances in Computing Research* (F.P. Preparata, ed.), vol. 2, JAI Press, Greenwich, CT, 1984, pp. 245–257.

Biography

XIAOYU SONG received the B.S. degree in Computer Science from the Changsha Institute of Technology, Changsha, China, in 1984, and received the "Laurea" and Ph.D degree in Computer Science from the University of Pisa, Italy, in 1986 and 1990. He is currently an Assistant Professor in the Department of Informatique et Recherche Operationelle at the University of Montreal, Canada. His current research interests include VLSI design and computation, algorithms, verification.