August 1988

UILU-ENG-88-2248
ACT-98

COORDINATED SCIENCE LABORATORY
College of Engineering
Applied Computation Theory

AN OPTIMUM CHANNEL ROUTING ALGORITHM IN THE RESTRICTED WIRE-OVERLAP MODEL

Dian Zhou

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-88-2248 (ACT-98)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION Semiconductor Research Corporation
6c. ADDRESS (City, State, and ZIP Code) 1101 W. Springfield Avenue Urbana, IL 61801	7b. ADDRESS (City, State, and ZIP Code) P.O. Box 12053 Research Triangle Park, NC 27709
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Semiconductor Research Corp. 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER SRC-87-DP-109PRE
8c. ADDRESS (City, State, and ZIP Code) P.O. Box 12053 Research Triangle Park, NC 27709	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
An Optimum Channel Routing Algorithm in the Restricted Wire Overlap Model

12. PERSONAL AUTHOR(S)
Zhou, Dian

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM ______ TO ______	August, 1988	18

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	VLSI, channel routing, algorithm, overlap model

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In this paper, we study the channel routing problem in the knock-knee mode with restricted edge overlap. We present an optimal algorithm which achieves $t = d_m$ with an overlap of $O(m)$ edges between any two nets, where t is the number of tracks, d_m is the density, and m is the multiplicity of the nets. The algorithm improves upon the previous results of Sarrafzadeh (1986) and has the properties: (1) it uses only vertical edge overlap, (2) it does not use any columns outside the channel, and (3) it connects adjacent terminals of a net by shortest wires. Furthermore, our algorithm reduces the overlap between two nets from $O(m^2)$ to $O(m)$. The running time of the algorithm is $O(d_m nm)$, where n is the number of nets.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

AN OPTIMUM CHANNEL ROUTING ALGORITHM IN THE RESTRICTED WIRE OVERLAP MODEL†

Dian Zhou

Coordinated Science Laboratory
1101 W. Springfield Avenue
University of Illinois, Urbana, IL 61801

Abstract

In this paper, we study the channel routing problem in the knock-knee mode with restricted edge overlap. We present an optimal algorithm which achieves $t = d_m$ with an overlap of $O(m)$ edges between any two nets, where t is the number of tracks, d_m is the density, and m is the multiplicity of the nets. The algorithm improves upon the previous results of Sarrafzadeh (1986) and has the properties: (1) it uses only vertical edge overlap, (2) it does not use any columns outside the channel, and (3) it connects adjacent terminals of a net by shortest wires. Furthermore, our algorithm reduces the overlap between two nets from $O(m^2)$ to $O(m)$. The running time of the algorithm is $O(d_m nm)$, where n is the number of nets.

† This work was partially supported by the Semiconductor Research Corporation under grant 87-DP-109.

I. Introduction

The essential reason for restricting wires from overlapping is the undesired capacitance introduced by the overlap of two wires. In any routing mode, Manhattan or Knock-knee, the capacitance between nets generally can not be avoided due to the crossing sections of wires. In addition, there is a coupling capacitance which exists between the parallel running wires (no overlap). However, wires running in parallel are allowed in the design of integrated circuits because the coupling capacitance introduced in this way is much smaller than the capacitance introduced by the overlap of two wires in previous technologies. As VLSI technology advances, the feature width of the wire has been scaled down into the submicron range and becomes nearly as small as the wire thickness [BM,ZPK,SM]. The vertical separation between two conducting layers becomes comparable to the horizontal separation between two adjacent wires in the same layer. Owing to the fringe effect, the coupling capacitance between two adjacent wires is comparable to that between two vertical overlap wires [ST,RB,D,DS,L,CA]. This situation is demonstrated in Figure 1. Consequently, whether wires run in parallel or in overlap will not make quite such a difference for the future VLSI design in terms of the capacitance introduced. This implies that the circuit performance will not be very much affected if restricted edge overlaps are introduced in routing.

In the rest of this paper, we will focus on the problem of using restricted edge overlap to reduce the channel width. We will also investigate the routings which produce shorter edge overlap for the same channel width. We first introduce the definition of the problem and point out the aspects in which the improvement can be made on the previously known results. Then, we present a new algorithm which is shown by a case studying method. Finally, in the discussion section we make some comments on the routing model discussed and on the wiring problem of the presented algorithm.

II. Definition of channel routing in the restricted edge overlap model

A channel routing problem CRP $\eta = \{ N_1, ..., N_n \}$ on a square grid with t tracks is as follows [S]: The horizontal line $y = 0$ is called the *Top* and the line $y = t + 1$ is called the *Bottom*. The *columns* are vertical lines located at positions $x = i$, $i = 1, 2, \cdots$. The leftmost column of the channel is called the left side, and the rightmost column is called the right side. A net N is a subset of $IN \times \{T, B\}$, where the first integer denotes the abscissa, and T and B stand for the Top and the Bottom, respectively. A net N_i with $|N_i| = k$ is called a *k-terminal* net. A CRP η is said to have *multiplicity* m if $|N_i| \le m$ for all i and $|N_i| = m$ for at least one m. The number of

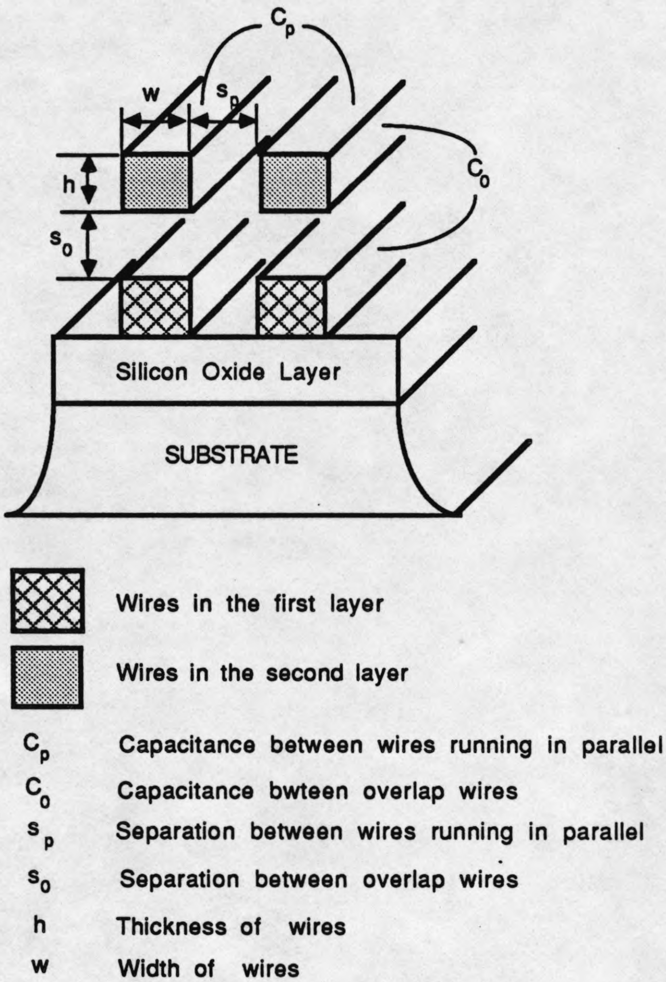


Figure 1

nets across a vertical line $x = x_0$, $x_0 \in (i, i+1)$ $(i=1,2,...)$, is defined as the density at $x = x_0$, which is denoted by $d(x_0)$. The density of a CRP η, denoted by d_m, is the maximum value of $d(x_0)$ over all columns.

It has been shown [S] that $t = d_m$ is an existential tight lower bound for channel width if $f(n,m)$ units (for any integer valued function $f(n,m)$) overlap between two nets is allowed. The algorithm of Sarrafzadeh achieves $t = d_m$ with edge overlap between any two nets less than $O(m^2)$. The extending strategy for routing "two-terminal nets" as well as the stair-like routing for connecting the terminals between the leftmost and the rightmost terminals of the net are used [S,MPS]. Two undesired features are therefore introduced. First, the area outside the two sides of the channel is used by those extending nets. In the worst case, a total $d_m/2$ extra columns is required. Secondly, the length of interconnection wires is relatively longer because some nets will be extended over their leftmost terminals. From the view point of VLSI design, using extra columns and longer interconnection wires are undesirable, since they make the global layout even more complicated [U] and make the delay of the signal propagation longer

[SPP,ZPK].

The new algorithm presented in this paper uses the restricted vertical edge overlap, and achieves the result $t = d_m$ with the following properties: No more than $O(m)$ edge overlap between any two nets, no extra columns (area) required, and shorter interconnection wires. The running time of this new algorithm is $O(d_m nm)$. Since $t = d_m$ is a lower bound on the channel width if no horizontal edge overlap is allowed, the upper bound on the channel width obtained is therefore tight. Furthermore, since the routing is completed inside the two sides of the channel, the area of the routing is also optimum.

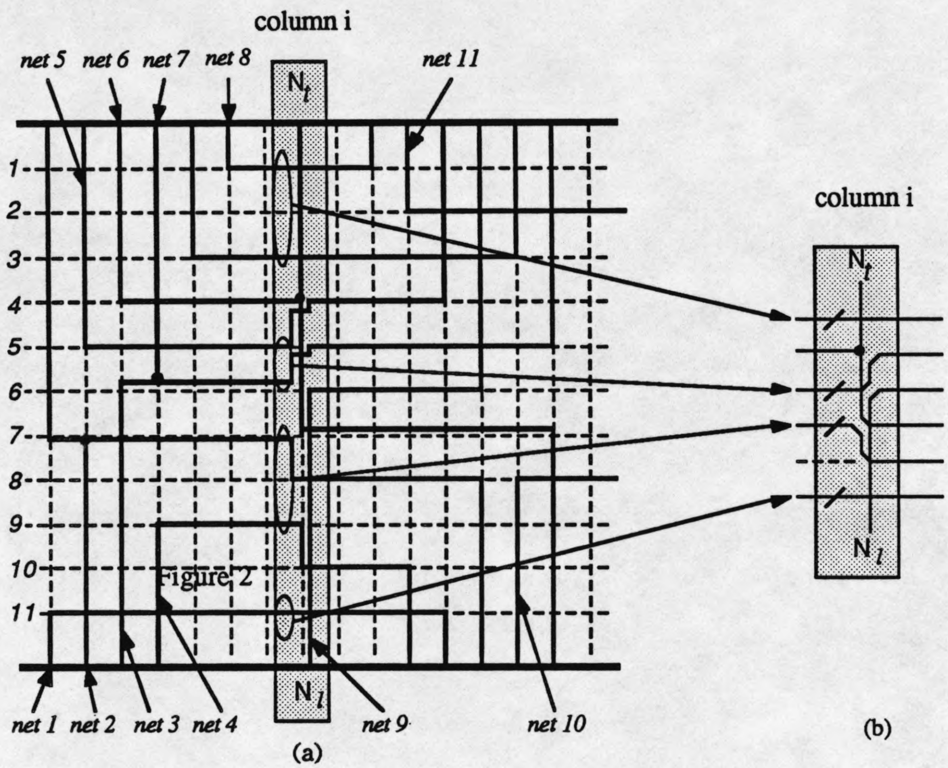
III. A new routing algorithm

The technique used in our algorithm is such that the nets across a column with their next terminals on the Top are always placed above the nets across this column with their next terminals on the Bottom. The algorithm constructs the routing in a column by column fashion. While processing column i, the intervals of interest are the open intervals $x \in (i-1, i)$ and $x \in (i, i+1)$. In this paper, we use $x = i_-$ to denote $(i-1, i)$ and $x = i_+$ to denote $(i, i+1)$. For instance, $d(i_+)$ is the density in $x \in (i, i+1)$. The algorithm starts from a chosen column i_0 with density $d(i_{0-}) = d_m$ and sweeps to the right side of the channel. Next, the algorithm starts from column i_0 and sweeps to the left side of the channel to complete the routing. Because there is no essential difference between operations in the right and left sweeps, we will only discuss the operations in the right sweep. Also, due to the column by column processing fashion, it is sufficient to show the routings of nets at an arbitrary interval $x \in (i-1, i+1)$.

The *direction* of a channel (or a net) is defined to be from the left to the right. The leftmost terminal of a net is called the *starting terminal*; the rightmost terminal is called the *ending terminal*; and the terminals between the starting and the ending terminals are called *continuing terminals*. We will use S, E and C to denote these three terminal states in a net, respectively. Let * be the empty element of a set. We introduce the following definition:

Column State: The column state is defined by a 6-tuple $(z_1,z_2,z_3,z_4,z_5,z_6)$, where z_1 and z_4, $z_1,z_4 \in \{T, B, *\}$, are the vertical positions (Top or Bottom) of two terminals at this column, z_2 and z_5, $z_2,z_5 \in \{S, C, E, *\}$, are these terminals' states in the nets they belong to, and z_3 and z_6, $z_3,z_6 \in \{T, B, *\}$, are the vertical positions (Top or Bottom) of next terminals of the nets with at least one of their terminals at this column.

In the example of Figure 2(a), the state of column i is (T, C, B, B, S, T), that means: (1) a net (net 6) has a continuing terminal on the Top at the considering column, and its next terminal is on the Bottom (the first three elements of the 6-tuple); (2) another net (net 9) has a starting terminal on the Bottom at this column, and its next terminal is on the Top (the last three elements of the 6-tuple). For easy reference, we have denoted the net with its terminal on the Top (Bottom) at the current column with N_t (N_b). A simplified expression for the routing structure of Figure 2(a) at interval $(i-1, i+1)$ (shaded area) is introduced in Figure 2(b), where a virgule on a wire denotes a set of parallel running wires.



Column state (C,T,B,S,B,T)

* is used to express a set of wires running in parallel. Therefore the routing structure in (a) for interval i-1 ≤ x ≤ i+1 is expressed by (b). If there are empty tracks between the parallel running wires, these empty tracks are assume to be kept in the same relative positions.

Figure 2

The nets intersecting a vertical line x are divided into two sets: $U(x)$ and $L(x)$. $U(x)$ consists of the nets whose closest terminal to the right of x is on the Top, and $L(x)$ consists of the nets whose closest terminal to the right of x is on the Bottom. The cardinalities of $U(x)$ and $L(x)$ are denoted by $NU(x)$ and $NL(x)$, respectively. The tracks intersecting the vertical line x are divided into two sets: $TU(x)$ and $TL(x)$ which consist of the tracks for routing the nets in $U(x)$ and the nets in $L(x)$, respectively. It is maintained in the algorithm that the tracks in $TU(x)$ are always above the tracks in $TL(x)$, and $|TU(i_+)| + |TL(i_+)| = d_m$. These definitions for the problem of Figure 2 at interval $(i-1,i+1)$ are illustrated in Figure 3. The reader should note the relations among the notations of routing structure at interval $(i-1,i+1)$ in Figure 2(a), Figure 2(b), and Figure 3. The graphic notation introduced in Figure 3 will be used throughout the rest of the paper.

Since the densitys $d(i_+)$ varies in general with i_+, it is important to choose correctly where to leave an empty track (either in $TU(i_+)$ or in $TL(i_+)$), whenever a net terminates at column i. (Remember the algorithm starts from column i_0 with $d(i_{0-}) = d_m$). The choice depends on the positions of the starting terminals to the right of column i. The empty tracks must be placed in the proper places so that the starting terminals following column i can be led into these empty tracks. A look-ahead method is used in the algorithm to determine the position of the empty track.

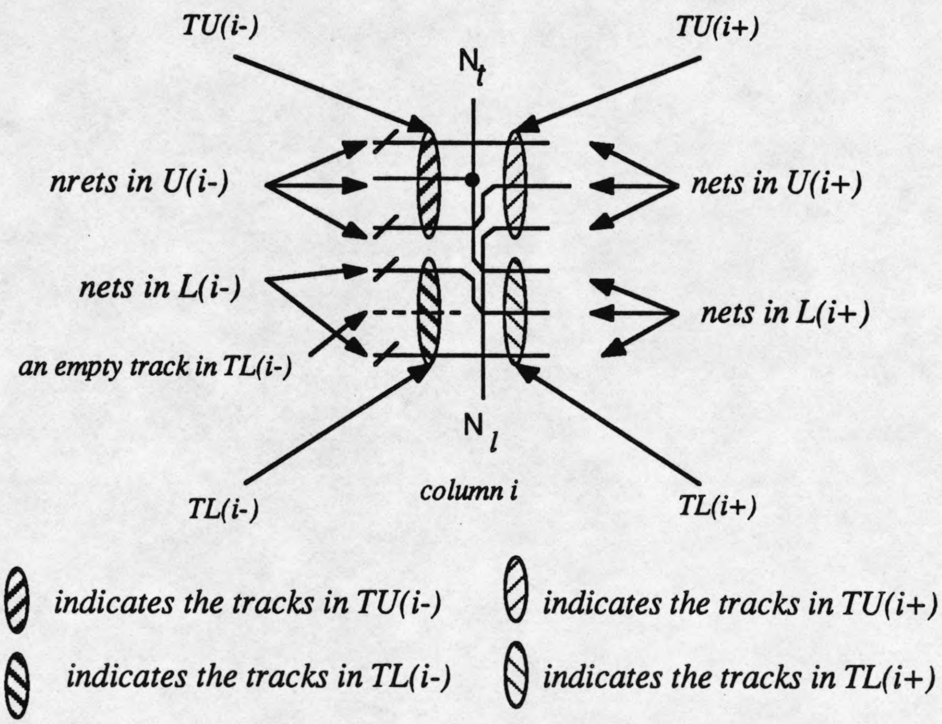


Figure 3

The algorithm uses the information of the vertical positions (Top or Bottom) of the next $(d_m - d(i_+))$ starting terminals following column i. A starting terminal is called the j*th starting−terminal* following column i if there are exactly $j-1$ starting-terminals located between column i and the column where this j*th* starting-terminal is. If there are two j*th* starting terminals at the same column, one of them, which is chosen randomly, will be considered as the j+1*th* starting terminal. Let set $NF(i_+)$ consist of all the j*th* starting-terminals following column i such that $j \le d_m - d(i_+)$. In these $|NF(i_+)|$ terminals, $NFT(i_+)$ of them are on the Top, and the remaining $NFB(i_+) = |NF(i_+)| - NFT(i_+)$ are on the Bottom. The algorithm leaves $NFT(i_+)$ empty tracks in $TU(i_+)$ and $NFB(i_+)$ empty tracks in $TL(i_+)$. This means that the empty tracks are always placed in the appropriate positions and the problem can be routed in d_m tracks.

As an example, all the notations introduced above for the problem of Figure 2 are as follows:

both $TU(i_-)$ and $TU(i_+)$ consist of tracks 1 through 6;

both $TL(i_-)$ and $TL(i_+)$ consist of tracks 7 through 11;

$NU(i_-) = 5$ (nets 3,5,6,7, and 8);

$NL(i_-) = 3$ (nets 1,2, and 4);

$NU(i_+) = 5$ (nets 3,5,7,8, and 9);

$NL(i_+) = 4$ (nets 1,2,4, and 6);

$NF(i_-) = 3$ (starting terminals of nets 9,10, and 11);

$NF(i_+) = 2$ (starting terminals of nets 10 and 11);

$NFU(i_-) = 1$ (starting terminal of net 11);

$NFL(i_-) = 2$ (starting terminals of nets 9 and 10);

$NFU(i_+) = 1$ (starting terminal of net 11);

$NFL(i_+) = 1$ (starting terminal of net 10).

It should be pointed out that the example in Figure 2(a) is a part of some CRP which is assumed to have the density $d_m = 11$. This is why we use 11 tracks in that figure.

As mentioned earlier that the algorithm starts right sweep from a column at which the density is the channel density, following initial conditions are hold consequently for the starting column i_0: $NL(i_{0-}) = |TL(i_{0-})|$, $NU(i_{0-}) = |TU(i_{0-})|$, $NFB(i_{0-}) = 0$, $NFT(i_{0-}) = 0$, and $|TU(i_{0-})| + |TL(i_{0-})| = d_m$. $d(i_{0-}) = d_m$. During the algorithm sweeps to the right, it keeps the following four invariants:

A1. The nets in $U(i_+)$ are always placed above the nets in $L(i_+)$.

A2. At the current column, any two nets overlap at most one vertical unit edge.

A3. $|TU(i_+)| - NU(i_+) \geq NFT(i_+)$, $|TL(i_+)| - NL(i_+) \geq NFB(i_+)$. This means that the empty tracks in $TU(i_+)$ and $TL(i_+)$ are always in the proper places to route the following starting terminals.

A4. $|TU(i_+)| + |TL(i_+)| = d_m$.

It can be checked that the invariants A1 through A4 are satisfied for the initial conditions at column i_o. We will study all possible states of a column and show how to route in each case to maintain the invariants. Generally, there are three classes of column states:

Class 1. There is only one terminal at the column.

Class 2. Two terminals at the column belong to the same net.

Class 2. Two terminals at the column belong to the different nets.

Since classes 1 and 2 are simpler than, or the special cases of class 3, we will discuss only the cases in class 3 where two terminals of the column belong to different nets. Omitting the symmetric cases, we have following fourteen column states: (T, S, T, B, S, B), (T, S, T, B, S, T), (T, S, B, B, S, T), (T, S, T, B, C, B), (T, S, T, B, C, T), (T, S, B, B, C, T), $(T, S, T, B, E, *)$, $(T, S, B, B, E, *)$, (T, C, T, B, C, B), (T, C, T, B, C, T), (T, C, B, B, C, T), $(T, C, T, B, E, *)$, $(T, C, B, B, E, *)$, and $(T, E, *, B, E, *)$. A set of pictures are drawn in Figure 4 to show the routing structures used in the algorithm for each column state. The necessary remarks have been noted in the captions to make it easy for reader to check the correctness of the invariants A1-A4.

1. Column State (T, S, T, B, S, B)

Use the routing structure shown in Figure 4(a) and the following relations hold:

$$NL(i_+) = NL(i_-) + 1,$$
$$NU(i_+) = NU(i_-) + 1,$$
$$NFB(i_+) = NFB(i_-) - 1,$$
$$NFT(i_+) = NFT(i_-) - 1,$$
$$TL(i_+) = TL(i_-),$$
$$TU(i_+) = TU(i_-),$$
$$|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.$$

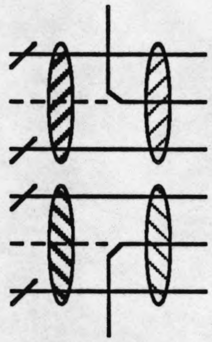


Figure 4(a).

2. Column State (T, S, T, B, S, T)

Use the routing structure shown in Figure 4(b) and the following relations hold:

$$NL(i_+) = NL(i_-),$$
$$NU(i_+) = NU(i_-) + 2,$$
$$NFB(i_+) = NFB(i_-) - 1,$$
$$NFT(i_+) = NFT(i_-) - 1,$$
$$TL(i_+) = TL(i_-) + 1,$$
$$TU(i_+) = TU(i_-) - 1,$$
$$|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.$$

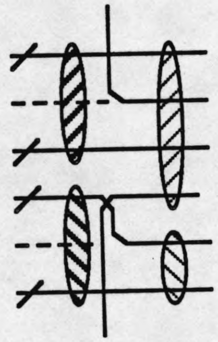


Figure 4(b)

3. Column State (T, S, B, B, S, T)

Use the routing structure shown in Figure 4(c) and the following relations hold:

$$NL(i_+) = NL(i_-) + 1,$$
$$NU(i_+) = NU(i_-) + 1,$$
$$NFB(i_+) = NFB(i_-) - 1,$$
$$NFT(i_+) = NFT(i_-) - 1,$$
$$TL(i_+) = TL(i_-),$$
$$TU(i_+) = TU(i_-),$$
$$|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.$$

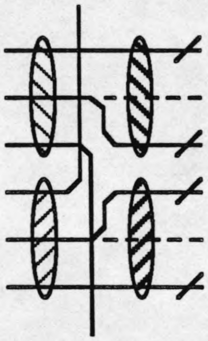


Figure 4(c)

4. Column State (T, S, T, B, C, B)

Use the routing structure shown in Figure 4(d) and the following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-),\\
&NU(i_+) = NU(i_-) + 1,\\
&NFB(i_+) = NFB(i_-),\\
&NFT(i_+) = NFT(i_-) - 1,\\
&TL(i_+) = TL(i_-),\\
&TU(i_+) = TU(i_-),\\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.
\end{aligned}
$$

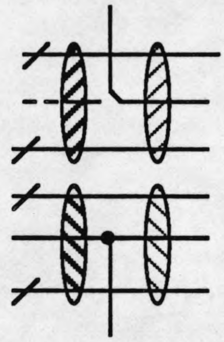


Figure 4(d)

5. Column State (T, S, T, B, C, T)

Use the routing structure shown in Figure 4(e) and the following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-) - 1,\\
&NU(i_+) = NU(i_-) + 2,\\
&NFB(i_+) = NFB(i_-),\\
&NFT(i_+) = NFT(i_-) - 1,\\
&TU(i_+) = TU(i_-) + 1,\\
&TL(i_+) = TL(i_-) - 1,\\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.
\end{aligned}
$$

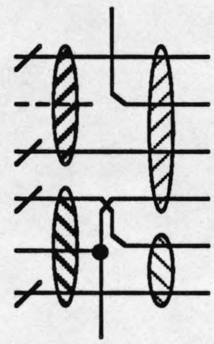


Figure 4(e)

6. Column State (T, S, B, B, C, T)

Use the routing structure shown in Figure 4(f) and the following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-),\\
&NU(i_+) = NU(i_-) + 1,\\
&NFB(i_+) = NFB(i_-),\\
&NFT(i_+) = NFT(i_-) - 1,\\
&TU(i_+) = TU(i_-),\\
&TL(i_+) = TL(i_-),\\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.
\end{aligned}
$$

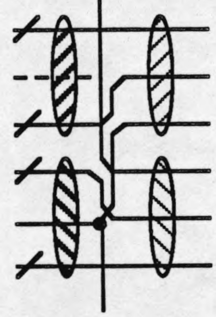
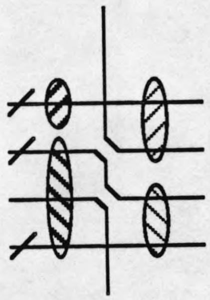


Figure 4(f)

7. Column State $(T, S, T, B, E, *)$

Use the routing structure shown in Figure 4(g). If $NFT(i_-) = 0$, or $NFT(i_-) \neq 0$ and the $(d_m - d(i_+))th$ starting-terminal following column i is on the Top, or there is no $(d_m - d(i_+))th$ starting-terminal following column i, route in structure (a) and the following relations hold:

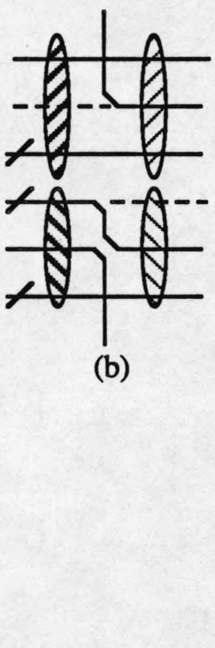
$$
\begin{aligned}
&NL(i_+) = NL(i_-) - 1, \\
&NU(i_+) = NU(i_-) + 1, \\
&NFB(i_+) = NFB(i_-), \\
&NFT(i_+) \leq NFT(i_-), \\
&TU(i_+) = TU(i_-) + 1, \\
&TL(i_+) = TL(i_-) - 1, \\
&t(i_+) \leq d(i_-) + NFB(i_-) + NFT(i_-) \leq d_m .
\end{aligned}
$$



(a)

If $NFT(i_-) \neq 0$ and the $(d_m - d(i_+))th$ starting-terminal following column i is on the Bottom, route in structure (b) and the following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-) - 1, \\
&NU(i_+) = NU(i_-) + 1, \\
&NFB(i_+) = NFB(i_-) + 1, \\
&NFT(i_+) = NFT(i_-) - 1, \\
&TU(i_+) = TU(i_-), \\
&TL(i_+) = TL(i_-), \\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m .
\end{aligned}
$$



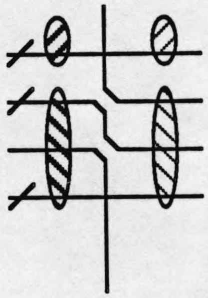
(b)

Figure 4(g)

8. Column State ($T, S, B, B, E, *$)

If $NFT(i_-) = 0$, or $NFT(i_-) \neq 0$ and the $(d_m - d(i_+))th$ starting-terminal following column i is on the Top, or there is no $(d_m - d(i_+))th$ starting-terminal following column i, route in structure (a) and the following relations hold:

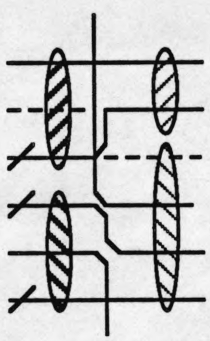
$$
\begin{aligned}
&NL(i_+) = NL(i_-) + 1, \\
&NU(i_+) = NU(i_-), \\
&NFB(i_+) = NFB(i_-), \\
&NFT(i_+) \leq NFT(i_-), \\
&TU(i_+) = TU(i_-), \\
&TL(i_+) = TL(i_-), \\
&t(i_+) \leq d(i_-) + NFB(i_-) + NFT(i_-) \leq d_m.
\end{aligned}
$$



(a)

If $NFT(i_-) \neq 0$ and $(d_m - d(i_+))th$ starting-terminal following column i is on the Bottom, route in structure (b) and the following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-) + 1, \\
&NU(i_+) = NU(i_-), \\
&NFB(i_+) = NFB(i_-), \\
&NFT(i_+) = NFT(i_-) - 1, \\
&TU(i_+) = TU(i_-) - 1, \\
&TL(i_+) = TL(i_-) + 1, \\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.
\end{aligned}
$$



(b)

Figure 4(h)

9. Column State (T, C, T, B, C, B)

Using the routing structure shown in Figure 4(i) and all parameters do not change.

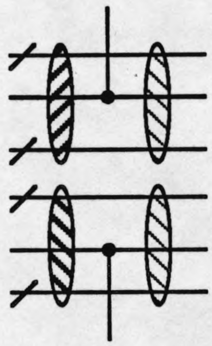


Figure 4(i)

10. Column State (T, C, T, B, C, T)

Use the routing structure shown in Figure 4(f) and the following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-) - 1, \\
&NU(i_+) = NU(i_-) + 1, \\
&NFB(i_+) = NFB(i_-), \\
&NFT(i_+) = NFT(i_-), \\
&TU(i_+) = TU(i_-) + 1, \\
&TL(i_+) = TL(i_-) - 1, \\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.
\end{aligned}
$$

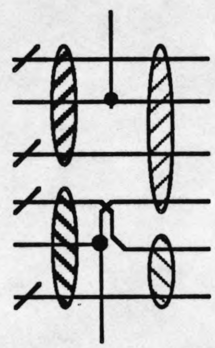


Figure 4(j)

11. Column State (T, C, B, B, C, T)

Use the routing structure shown in Figure 4(k) and the following relations hold:

$$\begin{aligned}
&NL(i_+) = NL(i_-),\\
&NU(i_+) = NU(i_-),\\
&NFB(i_+) = NFB(i_-),\\
&NFT(i_+) = NFT(i_-),\\
&TU(i_+) = TU(i_-),\\
&TL(i_+) = TL(i_-),\\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.
\end{aligned}$$

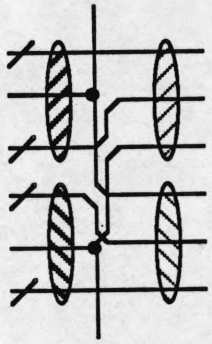
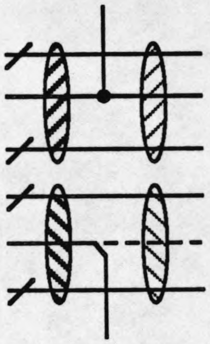


Figure 4(k)

12. Column State $(T, C, T, B, E, *)$

Route in the structure shown in Figure 4(i). If the $(d_m - d(i_+))th$ starting-terminal following column i is on the Bottom, or there is no $(d_m - d(i_+))th$ starting-terminal following column i, route in structure (a) and the following relations hold:

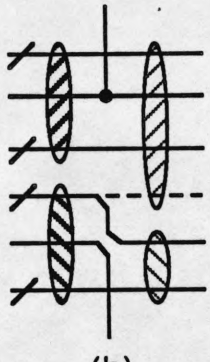
$$\begin{aligned}
&NL(i_+) = NL(i_-) - 1,\\
&NU(i_+) = NU(i_-),\\
&NFB(i_+) \le NFB(i_-) + 1,\\
&NFT(i_+) = NFT(i_-),\\
&TU(i_+) = TU(i_-),\\
&TL(i_+) = TL(i_-),\\
&t(i_+) \le d(i_-) + NFB(i_-) + NFT(i_-) \le d_m.
\end{aligned}$$



(a)

If the $(d_m - d(i_+))th$ starting-terminal following column i is on the Top, route in structure (b) and the following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-) - 1,\\
&NU(i_+) = NU(i_-),\\
&NFB(i_+) = NFB(i_-),\\
&NFT(i_+) = NFT(i_-) + 1,\\
&TU(i_+) = TU(i_-) + 1,\\
&TL(i_+) = TL(i_-) - 1,\\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.
\end{aligned}
$$



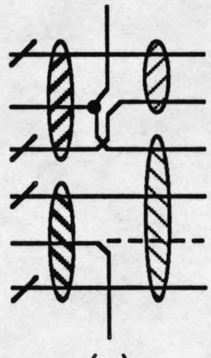
(b)

Figure 4(l)

13. Column State $(T, C, B, B, E, *)$

Route in the structure shown in Figure 4(m). If the $(d_m - d(i_+))th$ starting-terminal following column i is on the Bottom, or there is no $(d_m - d(i_+))th$ starting-terminals following column i, route in structure (a) and the following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-),\\
&NU(i_+) = NU(i_-) - 1,\\
&NFB(i_+) \le NFB(i_-) + 1,\\
&NFT(i_+) = NFT(i_-),\\
&TU(i_+) = TU(i_-) - 1,\\
&TL(i_+) = TL(i_-) + 1,\\
&t(i_+) \le d(i_-) + NFB(i_-) + NFT(i_-) \le d_m.
\end{aligned}
$$



(a)

If the $(d_m - d(i_+))th$ starting-terminal following column i is on the Top, route in structure (b) and the following relations hold:

$$NL(i_+) = NL(i_-),$$
$$NU(i_+) = NU(i_-) - 1,$$
$$NFB(i_+) = NFB(i_-),$$
$$NFT(i_+) = NFT(i_-) + 1,$$
$$TU(i_+) = TU(i_-),$$
$$TL(i_+) = TL(i_-),$$
$$|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m.$$

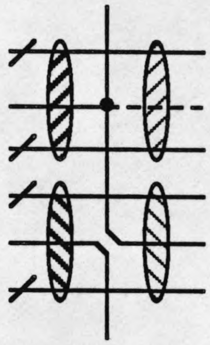
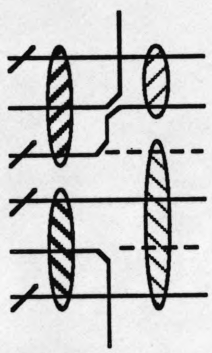

(b)

Figure 4(m)

14. Column State $(T, E, *, B, E, *)$

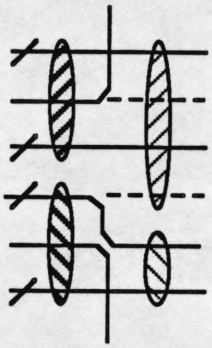
Route in the structure shown in Figure 4(n). If both $(d_m - d(i_+))th$ and $(d_m - d(i_+))th$ starting-terminals following column i are on the Bottom, route in structure (a) and the following relations hold:

$$NL(i_+) = NL(i_-) - 1,$$
$$NU(i_+) = NU(i_-) - 1,$$
$$NFB(i_+) = NFB(i_-) + 2,$$
$$NFT(i_+) = NFT(i_-),$$
$$TU(i_+) = TU(i_-) - 2,$$
$$TL(i_+) = TL(i_-) + 2,$$
$$t(i_+) \le d(i_-) + NFB(i_-) + NFT(i_-) \le d_m.$$


(a)

If both $(d_m - d(i_+))th$ and $(d_m - d(i_+))th$ starting-terminals following column i are on the Top, route in structure (b) and following relations hold:

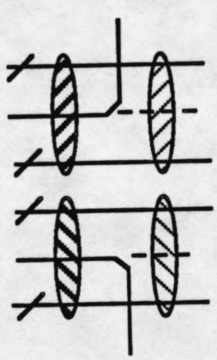
$$
\begin{aligned}
&NL(i_+) = NL(i_-) - 1, \\
&NU(i_+) = NU(i_-) - 1, \\
&NFB(i_+) = NFB(i_-), \\
&NFT(i_+) = NFT(i_-) + 2, \\
&TU(i_+) = TU(i_-) + 2, \\
&TL(i_+) = TL(i_-) - 2, \\
&t(i_+) \le d(i_-) + NFB(i_-) + NFT(i_-) \le d_m .
\end{aligned}
$$



(b)

Otherwise, route in structure (c) and following relations hold:

$$
\begin{aligned}
&NL(i_+) = NL(i_-) - 1, \\
&NU(i_+) = NU(i_-) - 1, \\
&NFB(i_+) \le NFB(i_-) + 1, \\
&NFT(i_+) \le NFT(i_-) + 1, \\
&TU(i_+) = TU(i_-), \\
&TL(i_+) = TL(i_-), \\
&|TU(i_+)| + |TL(i_+)| = |TU(i_-)| + TL(i_-)| = d_m .
\end{aligned}
$$



(c)

Figure 4(n)

We now analyz the properties of the presented algorithm. Since any two nets can overlap with each other only at the columns which contain the terminals of these two nets, and at each column there is at most one unit overlap, the total overlap between any two nets is at most $2m$ (or $O(m)$). From the algorithm described above, obviously, there is on extra columns outside the channel needed. Notice that to connect a terminal at $x = x_1$ on the lower shore to the next terminal of the same net at $x = x_2$ on the upper shore, the algorithm always uses a wire with length $d_m + |x_1 - x_2|$. This is the Manhattan distance between these two terminals. In this sense, the algorithm creats a routing with a short interconnection wires. Finally, since $O(d_m)$ time is needed for the operation of each column and there are at most nm non-empty columns, the running time of the algorithm is $O(d_m nm)$.

V. Discussion

We have shown that a Knock-Knee plus restricted overlap routing model is feasible in future VLSI design. A channel routing algorithm was presented based on this routing model, which is optimal in channel width and area used for routing. The edge overlap between any two nets has been reduced from the previous result of $O(m^2)$ to $O(m)$. An ideal feature of the algorithm is that it doesn't use any columns outside the channel sides. An open question would be the wiring problem of this routing model: Our algorithm uses at least four layers in general, are there possible algorithms which achieve $t = d_m$ and are three layer wirable?

References

[S] M. Sarrafzadeh, "Hierarchical approaches to circuit layout," Ph. D Thesis, 1986.

[BM] H. B. Bakoglu and J. D. Meindl, "Optimal interconnection circuits for VLSI," IEEE Trans. Electron Device, Vol. ED-32, No. 5, May 1985.

[SM] K. C. Saraswat and F. M. Mohammadi, "Effect of scaling of interconnections on the time delay of VLSI circuits," IEEE J. Solid State Circuits, Vol. SC-17, pp. 275-280, Apr. 1982.

[ST] T. Sakurai and K. Tamaru, "Simple Formulas for two- and three-dimensional capacitance," IEEE Trans. Electron Devices, Vol. ED-30, No. 2, Feb. 1983.

[RB] A. E. Ruehli and P. A. Brennan, IEEE J. Solid State Circuits, Vol. SC-8, No. 8, Aug. 1973.

[D] J. W. Duncan, IEEE Trans. Microwave Theory Tech., Vol. MTT-15, No. 10, Oct. 1967.

[DS] R. L. M. Dang and N. Shigyo, IEEE Electron Device Lett., Vol. EDL-2, No. 8, Aug. 1981.

[ZPK] D. Zhou, F. P. Preparata, and S. M. Kang, "Interconnection delay in very high-speed VLSI," ICCD '88 Proceedings.

[CA] J. Chilo and T. Arnaud, "Coupling effects in the time domain for an interconnecting Bus in high-speed GaAs logic circuits," IEEE Trans. Electron Devices, Vol. ED-31, No. 3, March 1984.

[L] E. T. Lewis, "An analysis of interconnect line capacitance and coupling for VLSI Circuits," Solid State Electronics, Vol. 27, Nos. 8/9, 1984.

[U] J. D. Ullman, "Computational Aspects of VLSI," Computer Science Press, Inc., 1984.

[MPS] K. Mehlhorn, F. P. Preparata, and M. Sarrafzadeh, "Channel Routing in Knock-Knee Mode: Simplified Algorithms and Proofs," Algorithmica (1986) 1: 213-221.