VLSI DESIGN
1998, Vol. 7, No. 1, pp. 73–84
Reprints available directly from the publisher
Photocopying permitted by license only

Minimum Crosstalk Vertical Layer Assignment for Three-Layer VHV Channel Routing

SHASHIDHAR THAKUR[a,*], KAI-YUAN CHAO[b] and D. F. WONG[c]

[a]*Department of Computer Sciences, University of Texas, Austin, Texas 78712, USA;*
[b]*Department of Electrical and Computer Engineering, University of Texas, Austin, Texas 78712, USA;*
[c]*Department of Computer Sciences, University of Texas, Austin, Texas 78712, USA*

With the increasing density of VLSI circuits, the interconnection wires are getting packed even closer. This has increased the effect of interaction between these wires on circuit performance and hence, the importance of controlling crosstalk. We consider the gridded channel routing problem where, specifically, the channel has 3 routing layers in the *VHV* configuration. Given a horizontal track assignment for the nets, we present an optimal algorithm for minimizing the crosstalk between vertical wiring segments in the channel by finding an optimal vertical layer assignment for them. We give an algorithm that minimizes total crosstalk between vertical wires on the same *V* layer on adjacent columns of the grid in $O(v \log v)$ time using $O(v)$ memory, where the channel has v columns. We then extend this algorithm to consider crosstalk between wires in non-adjacent columns and between wires on different layers. Finally, we show how our algorithms can be extended to take crosstalk tolerance specifications for nets into account.

Keywords: Channel routing, crosstalk, interconnect, signal integrity

1. INTRODUCTION

With VLSI fabrication entering the deep sub-micron era, devices and interconnection resources are being placed at an ever increasing proximity. Reduction in the interconnection and transistor switching delays results in faster signal transition times. All these factors increase the coupling effect (inductive and capacitive) between wiring resources. Increased coupling effect not only increases signal delays, but also decreases signal integrity due to transmission line behavior. This phenomenon is called *crosstalk*. In addition to the coupling effect, crosstalk depends on signal transition time [1]. We consider the minimization of coupling effect when the signal transition time is fixed at a value determined by the technology used.

There is some work in the literature on crosstalk minimization for multi-chip modules [2, 3, 4] and for integrated circuits [6]. In this paper, we address the crosstalk minimization problem during chan-

*Corresponding author.

nel routing of integrated circuits. Previous work on the crosstalk problem in channel routing falls into two main categories. In the first category, the gridless routing model is used. Spacings between wires are adjusted to reduce crosstalk [3]. In the second category, the gridded 2-layer channel model is used. Horizontal track permutations are done to minimize crosstalk [6]. In this paper, we consider the gridded 3-layer *VHV* model for the channel. In this model, the vertical wires may be assigned to either the top or bottom routing layers. We assume that track assignment for the middle *H* layer has been specified. This may be accomplished by using a conventional channel width minimization algorithm [11] or by permuting horizontal tracks to minimize crosstalk [6]. The only degree of freedom that is available now is to choose the layer assignments for the vertical wiring segments. We show how to do this assignment in order to minimize the total crosstalk between vertical wires.

In actual designs, the tolerance (noise budget) for crosstalk on a net is specified by the noise sensitivity of the net and by its wiring length (which is fixed for the proposed layer assignment problem). This can be translated into tolerance specifications for each vertical wire segment in the net (described in Section 4). We show how the total crosstalk between vertical wires can be minimized, while each wire satisfies its noise tolerance.

Our algorithms target the total vertical crosstalk only. The crosstalk between the horizontal wires is an important component of the total crosstalk in the channel. Since our algorithms take the horizontal track assignment as an input, one could use previously developed algorithms that target crosstalk while doing this assignment [6]. Thus the input to our algorithms can take the crosstalk between horizontal wires into account.

In the next section we introduce the notation to be used in this paper. The model for estimating crosstalk is formally specified. In section 3, we describe the algorithm that finds an optimal layer assignment for the vertical wires with the objective of minimizing the total crosstalk between them. In section 4 we show how the algorithm can be extended to allow for a specificaiton of crosstalk tolerance for nets.

2. PROBLEM DEFINITION

We first formalize the channel routing problem. A *channel* is a layered rectangular routing area with pins placed at the top and bottom edges. We assume that there is a grid superimposed over all the layers of the channel and that all terminals are on the grid points on the top and bottom edges of the channel. Such a channel is called a *gridded channel*. The *channel routing problem* is to connect pins in each net. A *valid routing solution* consists of a routing for wires along grid edges such that two wires do not share a grid edge or grid point. Thus, the following holds for a valid routing solution.

1. Wires may be routed only on grid edges. No two wires of different nets can share a common grid edge or grid point. We say that there is a *horizontal wiring violation* if two horizontal wiring segments share a grid edge or a grid point. Similarly, we say that there is a *vertical wiring violation* if two vertical wires share a grid edge or a grid point.
2. Each layer is reserved exclusively for horizontal or vertical wires. We call a layer reserved for horizontal wires a *horizontal layer*, which will be denoted *H*. Similarly, we call a layer reserved for vertical wires a *vertical layer*, which will be denoted *V*.

We assume that the routing channel has three layers in the *VHV* configuration i.e., the top and bottom layers are for vertical wires and the middle layer for horizontal wires. The number of tracks used in a channel routing solution is called the *channel width* and is denoted *w*. The number of nets crossing a column is called the *local density* of the column. The maximum of all local densities is called the *channel density* and is denoted *d*.

Obviously, the channel density is a lower bound on the channel width since there is only one horizontal layer.

For a 3 layer *VHV* channel there are no vertical conflicts. Thus, the left edge algorithm can find a valid routing solution with optimum channel width [7] i.e., channel width equals channel density. A optimum solution that does not use doglegs can be determined by this algorithm. *Hence, we assume that the input to the algorithms is a routing solution that does not use doglegs.*

Let the number of vertical columns in the channel grid be v i.e., the number of pins on each side of the channel is v. The pins are numbered 1, 2,..., v on the top and $v+1$, $v+2$,..., $2v$ at the bottom of the channel, both left to right. Thus, the channel is a $w \times v$ grid.

The input to the algorithm specifies the values of ω, v, the pins in the channel that are connected by each net and the horizontal track number used to route each net as determined by the channel routing algorithm. We denote the set of vertical wires by W. Each vertical wire is labeled by the number of the pin it is connected to. For a wire connecting the two terminals in the same column, it is labeled by the number of the pin at the top of the column. For other wires, this label is uniquely determined by its location. A *vertical layer assignment* of the vertical wires is mapping Y: $W \rightarrow \{T,B\}$. Here $Y(i)=T$ means that the vertical wire i is assigned to the top vertical layer. Similarly, $Y(i)=B$ means that the vertical wire i is assigned to the bottom vertical layer.

Consider a given routing solution specified by a horizontal track assignment and a layer assignment of the vertical wires. Let *distance* (i,j) denote the horizontal distance between two vertical wires, $i,j \in W$. Also, let *coupling_length* (i,j) be the number of horizontal grid lines in the channel for which there is no vertical wire between wires i and j. The *vertical crosstalk between two vertical wires* $i,j \in W$, $i \neq j$ and belonging to different multi-terminal nets, is defined as,

$$C_{i,j} = coupling\_length\,(i,j)/(distance(i,j))^s \quad (1)$$

where s is some constant (s is about 1.34 for capacitive coupling [10] but, due to inductive coupling, the total effective s is about 2 [8]). For two wires belonging to the same net $C_{i,j}=0$.

Thus, crosstalk between two vertical wires on the same layer, and belonging to distinct nets, is proportional to the lengths that are not shielded from each other by some other vertical wire. Across two layers, wires are unshielded from each other, and crosstalk is proportional to the overlapping lengths. It is difficult to accurately model the crosstalk across different layers, since different materials are filled between metal layers. A more accurate estimate for such wires can be used without requiring any modifications to our algorithms.

The above concepts are illustrated in the following example.

Example 1 Refer to the channel in Figure 1. In this figure, $v=5$ and $w=3$. Let $s=1$ and let the distance between the two V layers be 2 units and between adjacent columns in the grid be 1 unit. Consider the layer assignment, $Y(1)=Y(2)=Y(3)=Y(9)=T$ and $Y(5)=Y(6)=Y(8)=B$. Then, the crosstalk $C_{1,2}=1/1=1$ while $C_{1,3}=0$ since vertical wire 2 allows no coupling between wires 1 and 3. Also, $C_{2,9}=0$ because there is no coupling and $C_{6,9}=1/\sqrt{13}$ because the wires are on distinct layers and hence the distance between them is $\sqrt{3^2+2^2}$. Finally, $C_{8,9}=0$ as wires 8 and 9 belong to the same multi-terminal net.

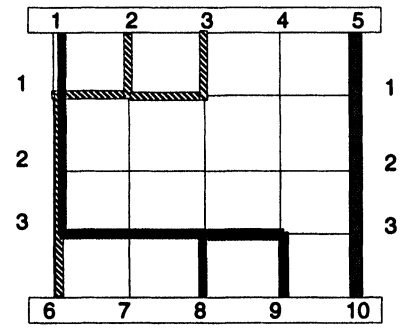


FIGURE 1 A sample channel.

The *total vertical crosstalk* in a routing solution is,

$$\sum_{i,j \in W, j \neq i} C_{i,j}$$

The problem we address in this paper is as follows:

Vertical Layer Assignment Problem *Given a 3 layer VHV channel specification, a set of multi terminal nets and a solution to the channel routing problem such that each net uses exactly one horizontal segment (no doglegs), find a layer assignment Y for the vertical wires that minimizes the total vertical crosstalk.*

An alternative objective function is defined in Section 4, for which a similar solution to the one we propose can be applied.

3. CROSSTALK MINIMIZATION ALGORITHM

We now describe the algorithms for minimizing vertical crosstalk. Our algorithms are intrinsically independent of the crosstalk model defined in the previous section. Thus any other model that assigns a value to $C_{i,j}$ statistically, i.e., the value depends only on the coordinates of the vertical wires and the horizontal track assignment, can be used in conjunction with our algorithms.

To simplify the presentation, we divide this section into three parts. In the first subsection we consider crosstalk between vertical wires on adjacent columns of the same V layer of the channel only. We show how to transform the crosstalk minimization problem, for this case, into one of the computing the shortest path in a graph. In the second subsection we generalize the solution to the case where we consider crosstalk between wires on the same layer that are at most K columns apart, for any specified K. In the last subsection, we extend the algorithm to take crosstalk between wires on different vertical layers into account.

3.1. Adjacent Columns Only

As mentioned before, in this section we consider crosstalk between vertical wires on adjacent columns on the same layer of the channel grid only. Thus, the vertical crosstalk $C_{i,j}$ is zero if $Y(i) \neq Y(j)$ or $|i-j| \neq 1$. We first show how to capture the channel and crosstalk information in terms of a directed graph. Then we prove that a shortest path computation on the graph yields an optimal solution to the crosstalk minimization problem.

We construct a graph $G(N, E)$ where the vertex set N has up to 4 vertices for each column of the channel C. Consider some column j of the grid. If column j has just one vertical wire then the possible layer assignments to that wire are T and B. We create two vertices, $(T)_j$ and $(B)_j$, corresponding to these two possibilities. If the column has two vertical wires then the possible layer assignments are represented by the ordered pairs (T, B), (B, T) (T, T) and (B, B), where the first entry represents the layer assignment to the vertical wire connected to the top pin in the column and the second entry represents the layer assignment for the vertical wire connected to the bottom pin. Corresponding to these possibilities, we generate vertices $(T, B)_j$, $(B, T)_j$, $(T, T)_j$, $(B, B)_j$. If the two vertical wires in column j have an overlap then only the vertices $(T, B)_j$ and $(B, T)_j$ are generated. This is because assigning these two wires to the same vertical layer leads to an infeasible routing solution. The above discussion is summarized in the following equations.

$$N_j = \begin{cases} \{(T)_j,\ (B)_j\} & \text{if column } j \text{ has only one vertical wire} \\ \{(B,T)_j,\ (T,B)_j\} & \text{if column } j \text{ has two overlapping vertical wires} \\ \{(T,B)_j,\ (B,T)_j,\ (T,T)_j,\ (B,B)_j\} & \text{if column } j \text{ has two non-overlapping vertical wires} \end{cases}$$

Let N_j be the set of vertices generated corresponding to column j of the grid. Then the vertex set is,

$$N = \{source,\ sink\} \bigcup_{j=1}^{v} N_j$$

Here *source* and *sink* are special vertices. We refer to the set N_j as the column j of N.

The edges are from each vertex in N_j to each vertex in N_{j+1}, for $j = 1, 2, \ldots, v-1$. The length of an edge equals the sum of vertical crosstalks between the vertical wires in column j and column $j+1$, for the vertical layer assignment specified by the two vertices linked by the edge. For example, the edge connecting $(B,\ T)_j$ and $(T,\ B)_{j+1}$ has length $C_{j,v+j+1} + C_{v+j,j+1}$. In addition, there is an edge from *source* to every vertex in N_1 and an edge from every vertex in N_v to *sink*. The lengths of these edges is 0.

Example 2 For the channel in Figure 1, the vertex subsets are, $N_1 = \{(B,T)_1,\ (T,B)_1\}$, $N_2 = \{(T)_2, (B)_2\}$, $N_3 = \{(B,T)_3,\ (T,B)_3,\ (T,T)_3,\ (B,B)_3\}$, $N_4 = \{(T)_4,\ (B)_4\}$ and $N_5 = \{(T)_5, (B)_5\}$. The corresponding graph in shown in Figure 2. The distance between adjacent columns in the channel grid is assumed to be 1 unit and $s = 1$.

Now the algorithm for layer assignment of vertical wires proceeds by finding the shortest directed path from *source* to *sink*. Denote this path by P. Such a path will have exactly one vertex from N_j for each $j = 1, 2, \ldots, v$. The layer assignment for the vertical wires is then derived from the labels of the vertices on the shortest path. The following pseudo-code shows the complete algorithm.

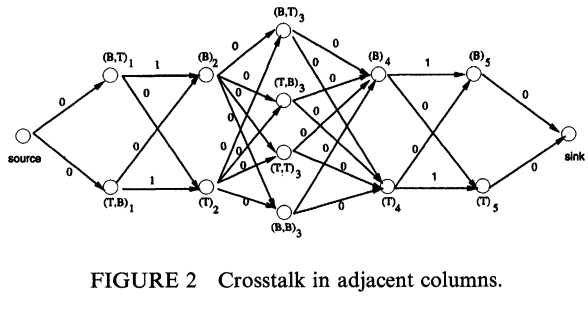


FIGURE 2 Crosstalk in adjacent columns.

Algorithm *assign_layers* (C)

```
/* Assign layers to the vertical wires in channel C*/
{
1.  Generate graph G(N, E) corresponding to C;
2.  Compute shortest path P from source to sink
    in G;
3.  for j = 1 to v
4.    if column j has only one vertical wire k then
5.      if (T)_j ∈ P then Y(k) = T else Y(k) = B;
6.    endif;
7.    if column j has two vertical wires then
8.      if (B, T)_j ∈ P then Y(j) = B, Y(j+v) = T;
9.      if (T, B)_j ∈ P then Y(j) = T, Y(j+v) = B;
10.     if (T, T)_j ∈ P then Y(j) = T, Y(j+v) = T;
11.     if (B, B)_j ∈ P then Y(j) = B, Y(j+v) = B;
12.   endif;
13. endfor;
}
```

Example 3 Continuing Example 2, in the graph in Figure 2 one shortest path is *source*, $(B,\ T)_1$, $(T)_2$, $(B,\ T)_3$, $(B)_4$, $(T)_5$, *sink* and has length 0. The optimal layer assignment is then, $Y(1) = Y(3) = B$ and $Y(2) = Y(5) = Y(6) = Y(8) = Y(9) = T$.

In general, for a column j, if $(B)_j \in P$ then the vertical wire in column j is assigned to the bottom vertical layer. Similar interpretations apply to vertices labeled by tuples.

The following theorem gives the optimality of the algorithm.

THEOREM 1 *If only the vertical crosstalk between vertical wires on the same V layer in adjacent columns of channel C is considered, then Algorithm assign-layers computes the layer assignment for vertical wires that minimizes the total vertical crosstalk in the channel C in time* $O(v \log v)$ *using* $O(v)$ *memory, where v is the number of columns in C.*

Proof Consider any vertical layer assignment to the vertical wires in channel C. It corresponds to choosing exactly one vertex from each N_j for $j = 1, 2, \ldots, v$. Thus, it defines a path from *source* to *sink* since every pair of vertices in consecutive columns of N is connected in $G(N, E)$. The length of this path equals the total vertical crosstalk in the

channel, under this layer assignment. Similarly, any path in G from *source* to *sink* contains exactly one vertex per column of N and gives a valid vertical layer assignment of the vertical wires in $\mathcal{C}$. The length of the path equals the corresponding total vertical crosstalk in $\mathcal{C}$. Hence, the shortest path in G corresponds to a minimum vertical crosstalk assignment to the vertical wires in $\mathcal{C}$.

We note that $|V| \leq 4v + 2$ and $|E| \leq 16(v-1) + 8$. The shortest path computation can be done in time $O(|V| \log |V| + |E|)$ [9] which reduces to $O(v \log v)$, for this problem. The rest of the steps can be done in time $O(v)$. The amount of memory used is proportional to the size of the graph and hence is $O(v)$.

3.2. Non-Adjacent Columns

In this section, we show how to extend Algorithm *assign_layers* to handle the case where we consider the crosstalk between vertical wires in non-adjacent columns of the channel. We assume that the crosstalk will be considered between vertical wires that are at a distance at most K, for some specified K. By choosing a large enough value for K one can obtain a layer assignment that minimizes the total vertical crosstalk to any given accuracy. We still consider crosstalk between wires on the same V layers only. Thus the vertical crosstalk $C_{i,j}$ is zero if $Y(i) \neq Y(j)$ or $|i-j| > K$.

To extend the formulation, we group the columns of the channel $\mathcal{C}$ into sets of K consecutive columns. Assume that $v = \alpha K$, for some $\alpha > 0$. Then the sets of consecutive columns are defined as $S_j = \{(j-1)K + i \mid i = 1, 2, \ldots K\}$ for $j = 1, 2, \ldots, \alpha$[1]. The construction of graph $G(N, E)$ is then quite similar to the one in the previous subsection. The difference is that the vertex subset N_j is constructed for each set of K consecutive columns, not for each column. The set N_j contains one vertex for each of the possible combinations of vertical layer assignments to the vertical wires in the columns in the set S_j for $j = 1, 2, \ldots, \alpha$. The vertices in N_j will be labeled by m-tuples for some $m \leq 2K$, since there can be up to $2K$ wires in K columns. These tuples represent the layer assignments to the wires in the K columns in S_j and the wires occur in the tuple in the order of increasing wire numbers.

Edges in G are from each vertex in N_j to each verex in N_{j+1} for $j = 1, 2, \ldots, \alpha$. The length of an edge is equal to the sum of vertical crosstalks between the vertical wires in the columns in S_j and the columns in S_{j+1}, under the layer assignments corresponding to the end-points of the edge. In addition, to take into account the crosstalk between vertical wires within the sets of columns S_j, we assign a cost to each vertex. The cost of a vertex in N_j equals the sum of crosstalks between the vertical wires in the columns in S_j, due to the layer assignment specified by the vertex. The *source* and *sink* vertices are added as before. They have a cost of 0 each. Edges of cost 0 are added from *source* to all vertices in N_1 and from all vertices in N_α to *sink*. Example 4 shows an instance of this construction.

Example 4 Consider the channel in Figure 3. In this figure, $v = 4$ and $w = 3$. Let $s = 1$, $K = 2$ and let the distance between adjacent columns in the grid be 1 unit. A illustration of the above graph construction for this channel is shown in Figure 4. Here $S_1 = \{1, 2\}$ and $S_2 = \{3, 4\}$. The correspond-

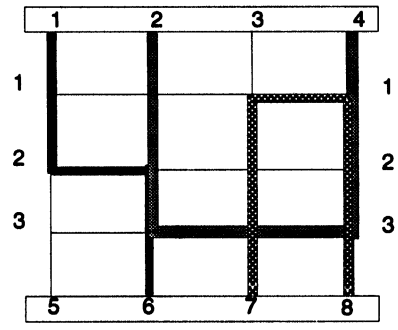
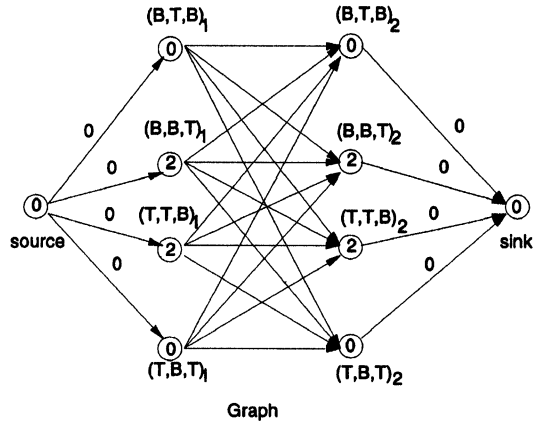


FIGURE 3 Another sample channel.

[1]If v is not a multiple of K then the last group of columns has less than K members.



	(B,T,B)$_2$	(B,B,T)$_2$	(T,T,B)$_2$	(T,B,T)$_2$
(B,T,B)$_1$	2.5	3.5	3	2.5
(B,B,T)$_1$	2.5	3	3	2
(T,T,B)$_1$	2	3	3	2.5
(T,B,T)$_1$	2.5	3	3.5	2.5

Lengths

FIGURE 4 Crosstalk in non-adjacent columns, $K=2$.

ing vertex subsets are $N_1=\{(B,T,B)_1, (B,B,T)_1, (T,T,B)_1, (T,B,T)_1\}$ and $N_2=\{(B,T,B)_2, (B,B,T)_2, (T,T,B)_2, (T,B,T)_2\}$. The matrix next to the graph gives the distances between the vertices in N_1 and N_2. The vertical wires, in the column group S_j, connected to the pins at the top of the channel first occur left to right in the tuple and then the vertical wires connected to the pins at the bottom of the channel are listed. Thus, the tuple $(B, T, B)_1$ corresponds to the assignment $Y(1)=B$, $Y(2)=T$ and $Y(6)=B$. We show examples of how the edge lengths and vertex costs are determined. Consider the vertex $(B, B, T)_1$. The cost of this vertex is $C_{1,2}+C_{1,6}=2/1+0=2$. For the edge from $(B, T, B)_1$ to $(B, T, B)_2$ the edge length is $C_{1,7}+C_{2,4}+C_{6,7}=1/2+0+2/1=2.5$. □

Having derived the graph $G(N, E)$, from the information in channel C, the algorithm now is to find a path P from *source* to *sink* that minimizes the sum of the lengths of the edges on P and the costs of vertices on P. A minor modification of the shortest path algorithm can be used to compute this path. The modification is to replace each vertex a (except *source* and *sink*) by a pair of vertices a_{in} and a_{out}. All the edges from a to some other vertex b are replaced by edges from a_{out} to b_{in}. An edge from a_{in} to a_{out} of length equal to the cost of a is added. Let the resultant graph be $G'(V', E')$. Now the regular shortest path algorithm can be applied to the transformed graph G' to get the minimum crosstalk layer assignment.

The modification of steps 3 through 13 in Algorithm *assign_layers*, to adapt to the fact that vertex labels can be m-tuples for some $m \leq 2K$, is straightforward.

Example 5 For the graph in Figure 4, one of the shortest paths in *source*, $(B, T, B)_1$, $(B, T, B)_2$, *sink* with length 2.5. The corresponding optimal layer assignment is $Y(1)=Y(6)=Y(7)=Y(8)=B$ and $Y(2)=Y(4)=T$.

The following theorem gives the optimality of the algorithm.

THEOREM 2 *If only the vertical crosstalk between vertical wires on the same V layer that are at a distance at most K in channel C is considered, then the above modification to Algorithm assign layers computes the layer assignment for vertical wires that minimizes the total vertical crosstalk in the channel C in time* $O(4^K \frac{v}{K}(4^K+\log v))$ *using* $O(\frac{v}{K}16^K)$ *memory.*

Proof The proof of optimality of this algorithm is quite similar to the one for Theorem 1. To show why the above memory and running times are correct, we observe that that the each column group S_j is represented by at most 4^K vertices in N_j. Since there are $\lceil \frac{v}{K} \rceil$ column groups we get $|V| \leq 4^K \lceil \frac{v}{K} \rceil$ and $|E| \leq 16^K \lceil \frac{v}{K} \rceil$. The transformation of the problem of finding a path with shortest sum of

vertex costs and edge lengths on G was transformed to the shortest path on G'. We note that $|V'| \leq 2|V|$ and $|E'| \leq |E| + |V|$. Since the computation of the shortest path can be done in time $O(|V'| \log |V'| + |E'|)$ we get the claimed result. □

In practical situations, one would not consider K more than 3 since the crosstalk contributed by distant columns drops almost quadratically due to the exponent s in the crosstalk function, as can be seen fom Equation 1. Also, due to the density of wires in the channel, wires at larger distances are likely to be shielded from each other. Hence, the complexity of the above algorithm as stated in Theorem 2 is acceptable.

3.3. Different V Layers

We now show how to take the crosstalk between wires on different V layers into account. Thus, now the vertical crosstalk $C_{i,j}$ is assumed to be zero only when $|i-j| > K$. This is a necessary extension of the previous subsections since grid spacings are of the same order as the distance between the two V layers (the inter-column spacing in the grid is about 0.8 μ and that between the two V layers is 2.8 μ in current technology [5]). Thus, crosstalk between wires in different layers is not negligible if no grounding plane exits between layers, which is the case with most VLSI designs. Furthermore, inter-layer crosstalk is getting significant due to deep sub-micron technology and increasing mutual capacitance of thick wires.

The algorithm in subsection 3.2 is extended to handle this case. The graph construction is identical to the one in the previous subsection. The vertex costs are modified by adding the crosstalk between vertical wires on different V layers for the layer assignment given by the label of the vertex. Similarly, the edge lengths are increased to reflect the crosstalk between wires on different V layers. As before, the path from *source* to *sink* that minimizes the sum of edge lengths and vertex costs on the path yields the minimum total vertical crosstalk layer assignment.

Example 6 For the channel in Example 4 let the distance between two columns of the grid be 1 unit and that between the two V layers be 2 units. To take into account crosstalk between wires in different V layers the cost of vertex $(B, T, B)_1$ has to be modified to $C_{1,2} + C_{2,6} = 2\sqrt{5} + 1/2 = 1.4$. Similar changes need to be made to other vertex costs. The length of the edge from $(B, T, B)_1$ to $(B, B, T)_2$ is increased by $C_{2,7} + C_{6,8} = 2/\sqrt{5} + 2/\sqrt{8} = 1.6$ and hence, the new length is 5.1. Other edge lengths are modified similarly. □

Similar arguments as before prove the optimality of the algorithm.

4. EXTENSION: ALTERNATIVE OBJECTIVE

During practical designs of high-speed VLSI circuits, crosstalk noise is usually treated as a constraint. In this section we show how the ideas in the previous sections can be extended to do the vertical layer assignment under crosstalk constraints on vertical wires. Typically, the tolerances are specified for entire nets, not individual wires. However, the noise tolerances of each vertical wires can be derived from the noise budget of a signal net by the following method: Given a noise budget of a net, the noise tolerance of each wire segment (both horizontal and vertical wire segments) is proportional to its length. Note that a long net can cross several channels, and crosstalk is accumulated over these channels. It is reasonable to use the above method to derive noise tolerance for each vertical wire segment, since the length of each horizontal and vertical wire segment constituting a net is fixed. The only degree of freedom is to assign vertical layers to each vertical wire segment, so as to minimize total vertical crosstalk in each channel, while the crosstalk on each wire segment is constrained by its tolerance. Recall that the crosstalk between horizontal wires has already been minimized in the horizontal track

assignment preceding the vertical layer assignment.

We can now define the modified vertical layer assignment problem precisely. Suppose a tolerance crosstalk for each vertical wire is specified by a function *tol:* $W \rightarrow \mathcal{R}$ such that, for a vertical wire $i \in W$, *tol* (i) gives the maximum permissible value of the crosstalk for the wire segment i. Define the crosstalk for wire $i \in W$ to be

$$\alpha(i) = \sum_{j \in W, j \neq i} C_{i,j}$$

Finding a valid layer assignment that adheres to these tolerances is an alternative objective. Thus the modified problem is:

Modified Vertical Layer Assignment Problem: *Given a* 3 *layer VHV channel specification, a set of multi terminal nets, a solution to the channel routing problem such that each net uses exactly one horizontal segment (no doglegs), and a tolerance function tol:* $W \rightarrow \mathcal{R}$. *Find a layer assignment Y for the vertical wires that minimizes the total vertical crosstalk such that, for each wire* $i \in W$, $\alpha(i) \leq tol(i)$.

A solution to this problem can be useful if nets are ordered by performance requirements and the crosstalk that each net can tolerate is different. A simple modification of the algorithm in subsection 3.1, as described below, finds a optimal layer assignment that keeps tht crosstalk at each vertical wire within the specified tolerance or declares the problem to be infeasible. We describe the modification for the case in Section 3.1 for the sake of simplicity. This can be easily extended to the cases in Sections 3.2 and 3.3.

Let $G(N, E)$ be the graph used to model the channel in Section 3.1. We observe that, for wire i, the value of $\alpha(i)$ is determined by the layer assignment to the wires in the columns to the immediate left and right of the column containing i. Thus, given a node corresponding to a particular layer assignment to the wires in the column containing i, and given a predecessor and a successor for it, we can determine exactly how much crosstalk $\alpha(i)$ is present on i under the layer assignment specified by the three nodes. Based on this we can transform the graph G into a new one, $G''(V'', E'')$ using two steps. The first step is as follows: Consider a vertex $x \in V - \{source, sink\}$ that represents a particular layer assignment in the column containing wire i. Let $V_1 \subset V$ be the set of predecessors and $V_2 \subset V$ the set of successors of x. Then we add $|V_1| + |V_2|$ copies of x to V'', one for each predecessor and each successor of x. We call the copy corresponding to $y \in V_1$ as x_y, and the copy corresponding to $z \in V_2$ as x_z. We assign a vertex cost to x_y equal to the length of arc (y, x) in G, and a cost equal to the length of arc (x, z) to x_z. The arc (x_y, x_z) with length 0 is added. This is done for each vertex in $V - \{source, sink\}$. Also, the edges (y_x, x_y) and (x_z, z_x) with length 0 are added to E''.

In the second step, the crosstalk $\alpha(i)$ due to layer assignments specified by vertices y, x and z is computed. It is easy to see this equals the sum of costs associated with the vertices x_y and x_z. If this sum exceeds *tol* (i) then the edge (x_y, x_z) is deleted from E''. This step is executed for each $x \in V - \{source, sink\}$.

Example 7 Consider the graph G as shown in Figure 2. The transformation described above is shown for a part of the graph G in Figure 5. The graph G is reproduced for easy reference. The vertex $(B)_2$ in G has two predecessors and four successors. Hence the block corresponding to this vertex has six vertices, one corresponding to each predecessor and each successor of $(B)_2$ in G. Consider the copy of $(B)_2$ that is adjacent to a copy of $(B, T)_1$. This vertex has cost 1, which was the cost of the edge $((B, T)_1, (B)_2)$ in G. The other vertex costs can be interpreted similarly.

Now any path from *source* to *sink* will contain two vertices, which are in fact copies of the same vertex in G, for each column in channel. Thus, the layer assignment for the vertical wires in a column is still uniquely determined by what vertices lie on the path. Since the crosstalk (length) of every edge in G gets duplicated in the copies of its source

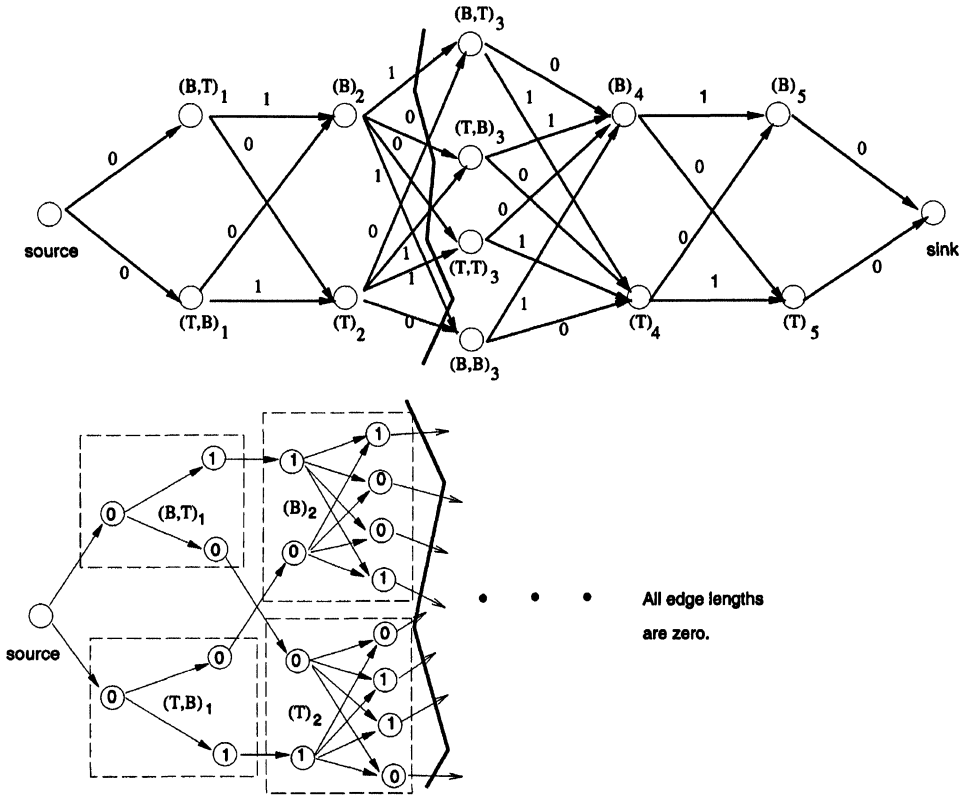


FIGURE 5 Crosstalk in adjacent columns with specified tolerance.

vertex and its destination vertex in G'', the length of the path now is double the crosstalk in the channel. Clearly, this does not affect the relative ordering of paths according to their lengths.

From the above discussion it can be seen that the solution to the modified crosstalk minimization problem is equivalent to finding the path in the transformed graph G'' that minimizes the total cost of vertices on it. Note that all the edges in G'' have zero weight. This path can be computed using the modification to the shortest path algorithm described in Section 3.2. If no path from *source* to *sink* exists in G'' then solving the problem under the given set of tolerances is infeasible. The complexity is the same as mentioned in Theorem 1 because $|V''| \leq 8v+2$ and $|E''| \leq |E|+16v \leq 32v-8$ i.e., $|V''|$ and $|E''|$ are both $O(v)$. We state the following result. It follows from the above discussion and Theorem 1.

THEOREM 3 *If only the vertical crosstalk between vertical wires on the same V lay in adjacent columns of channel C is considered, then the above modification to Algorithm assign_layers computes the optimal solution to the modified vertical layer assignment problem in the channel C in time O(v* log *v) using O(v), memory, where v is the number of columns in C.*

The method for dividing the tolerance for a net among the wires constituting it, described at the beginning of this section, is only a simple way of translating net tolerance to wire tolerance specifications. In general, this might result in a vertical layer assignment problem being declared infeasi-

ble, while it might have been feasible had the tolerances been divided in some other way. A remedy to this is to use the above algorithm as an inner loop in an iteration. In the event that the problem is declared infeasible, the tolerance for critical wires, i.e., those wires for which no corresponding edge exists in E'', can be reduced in the next iteration.

5. CONCLUSIONS

With increasing VLSI circuit densities, the problem of minimizing crosstalk between interconnection resources becomes important. We showed how to minimize the crosstalk between vertical wiring segments for a 3-layer *VHV* channel routing problem. Our algorithm is optimal and makes use of known efficient solutions to the shortest path problem. An extension of this method shows how to minimize the total vertical crosstalk while, at the same time, keeping the crosstalk on each vertical wire (or each net) within a specified bound.

Acknowledgements

This work was partially supported by the Texas Advanced Research Program under Grant No. 003658459 and by a DAC Design Automation Scholarship.

References

[1] Bakoglu, H. B. (1990). *Circuits, interconnections and packaging for VLSI*. Addison-Wesley.

[2] Chao, K.-Y. and Wong, D. F. (1994). Layer assignment for high-performance multi-chip modules. In *Proceedings of the International Conference on Computer Aided Design*, 680–685. IEEE/ACM.

[3] Chen, H. K. and Wong, C. K. (1992). Wiring and crosstalk avoidance in multi-chip module design. In *Proceedings of the Custom Integrated Circuits Conference*, 28.6.1–28.6.4. IEEE.

[4] Cho, J. D., Raje, S., Sarrafzadeh, M., Sriram, M. and Kang, S. M. (1993). Crosstalk-minimum layer assignment. In *Proceedings of the Custom Integrated Circuits Conference*, 29.7.1–29.7.4. IEEE.

[5] Denboer, A. (1994). Inside today's leading edge microprocessors. *Semiconductor International*, 64–66.

[6] Gao, T. and Liu, C. L. (1993). Minimum crosstalk channel routing. In *Proceedings of the International Conference on Computer Aided Design*, 692–696. IEEE/ACM.

[7] Hashimoto, A. and Stevens, S. (1971). Wire routing by optimizing channel assignment within large apertures. In *Proceedings of the Design Automation Conference*, 155–169. IEEE/ACM.

[8] Johnson, H. W. and Graham, M. (1993). *High-Speed Digital Design*. Prentice Hall Inc.

[9] Papadimitriou, C. and Steiglitz, K. (1982). *Combinatorial Optimization – Algorithms and Complexity*. Prentice-Hall Inc.

[10] Sakurai, T. and Tamaru, K. (1983). Simple formulas for two and the three dimensional capacitance. *IEEE Transactions on Electronic Devices*, **70**, 183–185.

[11] Yoshimura, T. and Kuh, E. S. (1982). Efficient algorithms for channel routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **1**, 25–35.

Authors' Biographies

Shashidhar Thakur received his B.Tech. in computer science from the Indian Institute of Technology, Bombay, India in 1990. He received the M.S. degree in computer sciences from the University of Texas at Austin in 1993 and is currently a Ph.D. candidate there. His research interests include algorithms, logic synthesis, FPGA architecture and synthesis, and routing. He has been a summer intern at AT & T Bell Labs. and Synopsys Inc.

Kai-Yuan Chao received the B.S. degree in Nuclear Engineering from the National Tsing Hua University, Taiwan in 1986. He received the M.S. degree in Medical Engineering from the National Yang-Ming Medical College, Taiwan, in 1988. He received the M.S.E. and Ph.D. degrees in Electrical and Computer Engineering from the University of Texas at Austin in 1992 and 1995, respectively. He is presently a Sr. CAD Engineer in at Intel Corporation, Hillsboro, OR. His current research interests are in physical design of VLSI, signal integrity, power optimization, and packaging. He is a member of Tau Beta Pi and IEEE.

D. F. Wong received the B.Sc. degree in mathematics from the University of Toronto and the M.S. degree in mathematics from the University of Illinois at Urbana-Champaign. He obtained

the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 1987. He is currently an Associate Professor of Computer Sciences at the University of Texas at Austin. Dr. Wong's main research interest is CAD of VLSI and has published more than 90 technical papers in this area. He is a coauthor of "Simulated Annealing for VLSI Design" (Kluwer Academic Publishers, 1988). At the 1986 ACM/IEEE Design Automation Conference, he received a best paper award for his work on floor-plan design. Dr. Wong is currently serving on the editorial board of IEEE Transactions on Computers.