International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 6, November-December 2014 ISSN 2278-6856

Two Algorithms for Minimizing Crosstalk in Two-Layer Channel Routing

Achira Pal[1], Tarak Nath Mandal[2], Abhinandan Khan[3], Rajat Kumar Pal[3], Alak Kumar Datta[4], and Atal Chaudhuri[1]

[1]Department of Computer Science and Engineering, Jadavpur University
188, Raja S. C. Mullick Road, Kolkata – 700 032, India

[2]SAP ABAP Consultant
IBM Global Business Services
IBM India Pvt. Ltd.

[2]Department of Computer Science and Engineering, University of Calcutta
Acharya Prafulla Chandra Roy Siksha Prangan
JD – 2, Sector – III, Saltlake City
Kolkata – 700 098, West Bengal, India

[3]Department of Computer and System Sciences
Visva-Bharati, Santiniketan – 731 235, India

Abstract

With the advancement of fabrication technology, devices and interconnecting wires are being placed in closer proximity and circuits are operating at higher frequencies. This results in crosstalk between overlapping wire segments. Work on routing channels with reduced crosstalk is a very important area for current research. The crosstalk minimization problem in the reserved two-layer Manhattan routing model is NP-complete, even if the channel instances are without any vertical constraints. The problem of crosstalk minimization remains NP-complete for general instances of channel specifications with both horizontal and vertical constraints. In this paper we have developed two algorithms for computing reduced crosstalk routing solutions on a given routing solution of minimum area for general instances of channel specifications. Performance of our algorithms is encouraging enough for most of the existing benchmark channels, and reduction in crosstalk for these channels is up to 28.34% for a given routing solution.

Keywords:- Algorithms, Channel routing problem, Crosstalk minimization, High performance routing, NP-completeness.

1. INTRODUCTION

In VLSI layout design it is required to realize a specified interconnection among different modules using minimum possible area. This is known as the routing problem. There exist several routing strategies for efficient interconnection among different modules. One of the most important types of routing strategies is channel routing [16], [31]. A channel is a rectangular (routing) region that has two open ends on the left and the right (that may or may not have any terminal to be assigned, and if it is so they are not fixed before assignment), and the other two (opposite) sides have two rows of fixed terminals (of equal length). A set of terminals that need to be electrically connected together is called a net. The terminals of the same net are assigned the same number. The terminals not to be connected are assigned the number zero. Throughout the paper we consider reserved layer Manhattan routing where only horizontal and vertical wire segments are assigned to respective layers for interconnecting the nets. Consider the case of long overlapping of wire segments on adjacent layers. Due to the overlap, there is a possibility of signal interference that may cause electrical hazards. In order to avoid this problem and to achieve feasible routing solutions for most of the channel instances, we assume that a layer has only horizontal wire segments or only vertical wire segments in a reserved layer model. The connection between a horizontal and a vertical wire segment of same net in adjacent layers is achieved using a via. This framework for routing a region is known as the reserved two-layer Manhattan routing model. Note that the problem of area minimization is the most important cost optimization criterion in routing a channel [4], [12], [14], [19], [20] followed by wire length minimization [10], [16], [18]. In order to minimize the routing area, the horizontal wire segments of the nets need to be distributed amongst a minimum number of tracks which is guided by two important constraints, viz., the horizontal constraints and the vertical constraints. These constraints can be characterized by two graphs, viz., the horizontal constraint graph (HCG) and the vertical constraint graph (VCG), respectively [16], [17], [31]. The local density of a column is the number of nets passing through the column. The channel density is the maximum number of nets passing through a column. We denote channel density by d_{max}. Throughout the paper we represent horizontal constraints using the complement of the HCG. We call the complement of the HCG, HC = (V, E), the horizontal non-

constraint graph (HNCG) and we denote it by HNC = (V, E′), where V is the set of vertices corresponding to the intervals and E′ = {{v_i, v_j} | {v_i, v_j} ∉ E}. The notation of the HNCG is introduced in [16], [17] to represent horizontal constraints. Note that a clique of the HNC corresponds to a set of non-overlapping intervals that may safely be assigned to the same track in a routing solution. The VCG, VC = (V, A) is constructed to represent the vertical constraints. For an acyclic VCG, we denote the length of the longest path in the VCG by v_{max}, where v_{max} is equal to the number of vertices belonging to the path. The channel routing problem (CRP) is the problem of assigning the horizontal wire segments of a given set of nets to tracks obeying horizontal and vertical constraints, so that the number of tracks required (and hence the channel area) is minimized. We say that a routing solution is feasible if all the nets are assigned without any violation. In this paper we consider the crosstalk minimization problem as a high performance prerequisite for channel routing in VLSI. As fabrication technology advances, devices and interconnects are placed in closer proximity and circuits operate at higher frequencies. This results in crosstalk between wire segments that is proportional to the coupling capacitance, which, in turn, is proportional to the coupling length, i.e., the total length of the overlap between wires [7], [23], [28], [29]. Crosstalk is also proportional to the frequency of operation and inversely proportional to the separating distance between wires. Therefore, it is important that these aspects be considered in testing the performance of a routing solution and the design of a performance driven routing algorithm. The aim should be to avoid long overlapping of wire segments and/or the wire segments that lie close to each other on the same layer. Work on routing channels with minimized crosstalk is a very important area for current research [2], [15], [16], [26]. It is desirable to design channel routing algorithms that consider this factor. The main objective in performance driven routing is to reduce signal delays due to crosstalk or distortion of information due to crosstalk. It may be noted that the crosstalk minimization problem in the reserved two-layer Manhattan routing model is NP-hard, even if the channels are without any vertical constraints [13], [15], [26]. Since the problem of minimizing crosstalk is NP-hard, several polynomial time heuristic algorithms have been proposed for such instances of channel specifications having no vertical constraints and have obtained outstanding results in terms of reduced crosstalk routing solutions [15], [26]. We may further mention that the problem of minimizing crosstalk for general instances of channel routing remains NP-hard even if vertical constraints are there in a channel [13]. In this paper we have developed algorithms for computing reduced crosstalk routing solutions on given routing solutions of minimum area for general instances of channel specifications. Performance of our algorithms is encouraging enough for most of the existing benchmark channels, and reduction on crosstalk for these channels is up to 28.34% on a given routing solution computed in [16]. This paper is organised as follows. In Section 2, a brief literature survey on the crosstalk minimization problem has been presented. We have discussed the issues relating to the said problem in Section 3. In Section 4, we have developed algorithms for computing reduced crosstalk routing solutions for general instances of channel specifications. Performance of our algorithms has been included in Section 5. The paper concludes in Section 6 with a few remarks and probable open directions for future study / investigations.

2. BACKGROUND

In this section we review some of the recent developments in relation to crosstalk minimization, mainly from the viewpoint of analog. The technique adopted is at times for network-on-chip (NoC) methodology and at other times it is for VLSI circuits designed based on CMOS technology. Sankaran and Katkoori [23] have proposed a simulated annealing (SA) based high-level synthesis algorithm for crosstalk activity minimization for a given data environment. Manem and Rose [9] have analysed the crosstalk produced in sub-lithographic programmable logic array (PLA) architectures and have proposed an alternative layout scheme to reduce the effects of crosstalk in adjacent wires. Reis *et al.* [22] have proposed a nonlinear crosstalk minimization algorithm that simultaneously considers other issues as well as those that are experimentally assessed. An efficient crosstalk delay reduction fault tolerant data bus encoding technique has been proposed in [24] which can reduce the total crosstalk delay by around 0.3% to 40% compared to other techniques for 12-bit to 71-bit data buses. Pande *et al.* [21] have proposed to lower the communication energy, which in turn may help to decrease the overall energy dissipation, by incorporating crosstalk avoidance coding in NoC data streams and by organizing the crosstalk avoidance code encoded data packets in an efficient manner such that the total number of encoding / decoding operations can be reduced over the communication channel. The crosstalk effects in mixed-signal ICs in deep submicron digital CMOS technology have been dealt by Liberali, Rossi and Torelli [6]. Their work illustrates the crosstalk phenomenon and its impact on the design of mixed analog / digital circuits with high accuracy specifications. Techniques to reduce analog / digital crosstalk have also been reviewed and discussed in this paper. Reduction of interconnect delay and interconnect power has become a primary design challenge in recent CMOS technology generations. In this regard, spacing between wires needs to be modified so that line-to-line capacitances can be optimized for minimal power under timing constraints. This has been approached by Moiseev, Wimer, and Kolodny [11] and they have presented a new algorithm for concurrent multi-layer interconnect spacing

that minimizes the total dynamic power dissipation caused by routing, while satisfying maximum delay constraints. Two problems have been introduced by Xu *et al.* [30], namely, couplings delay deterioration and crosstalk. This paper presents a timing-driven global routing algorithm along with consideration of coupling effects and crosstalk avoidance. A frequency-domain approach to efficiently simulate and minimize the crosstalk between high speed interconnects have been proposed by Ernesto Rayas-Sánchez [1]. The author has claimed that the simulation method proposed, yields good accuracy. In this context, Maheswari and Seetharaman [8] have proposed an energy-efficient error control code for the on-chip interconnection link capable of correcting any type of error patterns including random and burst errors. The proposed code provides crosstalk avoidance by reducing the coupling capacitance of the interconnecting wires. With the exponential reduction in the scaling of feature size, inter-wire coupling capacitance becomes the dominant part of load capacitance.

3. CROSSTALK MINIMIZATION PROBLEM

Channel routing has been used extensively in the layout of integrated chips in the last four or more decades. The CRP of area minimization is an NP-complete problem [5], [16], [25], [27]; several heuristics have been proposed for routing channels in different routing models [4], [12], [14], [16], [19], [20], [31]. The problem is polynomial time computable if the instances are free from any vertical constraints and we are interested only in resolving horizontal constraints in the two-layer VH channel routing model [3], [16], [18]. The same algorithm is applicable in computing routing solutions for any channel instances in the $V_{i+1}H_i$ routing models with alternating vertical and horizontal layers for any value of $i \geq 1$ [16].

Since the problem of minimizing area for the instances of routing channels without any vertical constraint is polynomial time solvable using only d_{max} tracks, such instances are termed as the simple channel specifications [13], [15], [26]. Hashimoto and Stevens [3] proposed a scheme for solving this problem, and according to Schaper [25], it can be implemented in $O(n(\log n + d_{max}))$ time, where d_{max} is the channel density and n is the number of nets belonging to the channel. Later on, Pal et al. [18] developed and analysed two different minimum clique cover based algorithms, MCC1 and MCC2, based on the scheme developed by Hashimoto and Stevens [3]; these algorithms have also been included in [16]. The first algorithm MCC1 is based on a graph theoretic approach that runs in O(n + e) time, where n is the number of nets and e is the size of the HNCG. The second algorithm, MCC2 is achieved using a balanced binary search tree data structure that takes O(nlogn) time for a channel with n nets [16], [18]. Though routing solution of only d_{max} tracks is guaranteed for the simple channel instances in the stated routing models, it may not be a good routing solution from the resulting crosstalk point of view.

We now discuss the presence of crosstalk between nets (or intervals) assigned to different tracks in a two-layer channel without any vertical constraint. If two intervals do not overlap, there is no horizontal constraint between the nets. That is, if there is a horizontal constraint between a pair of nets, there is a possibility of having a *measurable* crosstalk between them. We quantify crosstalk in terms of number of units a pair of nets overlaps on adjacent tracks in a feasible routing solution.

Consider the problem of minimizing crosstalk in a two-layer VH routing model. Suppose we have three intervals *a*, *b*, and *c* as shown in Figure 1(a), in a feasible routing solution of three tracks only.

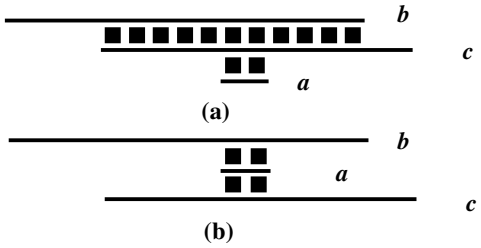

Figure 1 Crosstalk minimization problem in two-layer VH channel routing, in the absence of vertical constraints. **(a)** A feasible three-track routing solution with three intervals of three different nets *a*, *b*, and *c* that are overlapping to each other. Nets *b* and *c* share 11 units of horizontal span in the channel (as they are assigned to two adjacent tracks), and nets *c* and *a* share 2 units, amounting a total of 13 units' cross coupling length. **(b)** Another feasible three-track routing solution for the same channel instance, with a total net sharing of 4 units of horizontal span; hence a minimized crosstalk routing solution is obtained just by reassigning the nets to tracks.

Since all the three nets, *a*, *b*, and *c* overlap, we are compelled to assign them to three different tracks on the same horizontal layer in any feasible routing solution. But the most interesting feature we can point out is that in Figure 1(a), nets, *b* and *c* share 11 units of horizontal span in the channel, and nets, *c* and *a* share 2 units, whereas in Figure 1(b), we have a net sharing of 4 units of horizontal span in total just by reassigning the nets to tracks. It is inevitable that the assignment of nets to tracks in Figure 1(b) produces a reduced crosstalk routing solution; in fact it is the minimum crosstalk three-track routing solution for this instance. We now consider the presence of vertical constraint in a channel and the situation evolved, due to this constraint. Suppose there is a vertical constraint (*c*, *a*) as shown in Figure 2(a), as in some column we have a terminal of net, *c* on the top row and a terminal of net, *a* at the bottom row. In this case, we cannot alter the sequence of assigning the nets, *c* and *a*, in order to compute a reduced crosstalk feasible routing solution, as we did in the case of Figure 1. Rather in any feasible routing solution of this instance, we have to assign the

interval of net, *c* to a track above the track to which the interval of net, *a*, is assigned. In this case in order to minimize crosstalk we can alter the assignment of the interval of net, *b* to any of the three tracks maintaining the vertical constraint. In fact a minimum crosstalk feasible routing solution of this example is shown in Figure 2(b). In this context we are interested to pose the following problem.

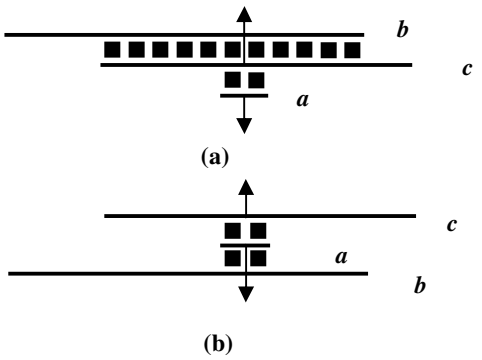

Figure 2 Crosstalk minimization problem in two-layer VH channel routing, in the presence of vertical constraints. **(a)** A feasible routing solution with a vertical constraint (c, a). **(b)** A reduced crosstalk routing solution considering the vertical constraint.

Problem: Crosstalk minimization in two-layer VH channel routing.

Instance: A simple channel specification (without any vertical constraint) and a positive integer, k.

Question: Is there a minimum area two-layer VH routing solution with the total crosstalk equal to k or less?

This problem has been proved NP-hard in [13]. As a result, several heuristic algorithms have been developed to compute reduced crosstalk routing solutions for the existing routing solutions of minimum area for the channel instances without any vertical constraints [15], [26]. The problem of crosstalk minimization remains NP-hard even if vertical constraints are there for general instances of CRP [13]. This proof is eventually followed by the proof by restriction, as the simple channel instances are the restricted versions of the general instances of channel specifications. Here the problem of crosstalk minimization is more restricted by the constraints and the nets are less flexible to move along the width (or height) of the channel, as the solutions have to obey both horizontal as well as vertical constraints. In this paper we have developed algorithms for computing reduced crosstalk routing solutions for a given routing solution of minimum area for general instances of CRP.

4. METHODS

Since the crosstalk minimization problem in two-layer channel routing, with and without vertical constraints, is NP-hard, it is unlikely that there exist polynomial time algorithms for minimizing crosstalk. In this paper we have developed two heuristic algorithms for computing reduced crosstalk routing solutions on the given routing solutions of minimum area for the general instances of CRP. Since the area minimization problem in channel routing is one of the most important cost optimization criteria in the physical synthesis of VLSI design, we start with a two-layer feasible routing solution, *S* of *t* tracks and compute another feasible routing solution, *S′* of the same *t* tracks with reduced total crosstalk. Note that for any feasible two-layer VH routing solution, *S* we can compute another two-layer VH routing solution, *S′* with total amount of crosstalk equal to zero. In computing *S′* we may introduce *t*–1 blank tracks into *S*, where between each pair of adjacent tracks in *S*, a blank track is introduced. As a result, *S′* becomes free from any crosstalk maintaining the geometry of the assumed routing model. So *S′* is a valid routing solution of nearly 2*t* tracks, without any crosstalk. Here the main point of emphasis is that if sufficient space is provided between the wires assigned to adjacent tracks (or layers), the amount of crosstalk should eventually reduce. Nonetheless, due to the fact that the area minimization problem is the most important cost optimization problem in the physical synthesis of VLSI design, we must not encourage computation of *S′* that uses almost twice the area of *S*. So it should be a trade-off between routing area and the resulting crosstalk in routing a channel. As a result, instead of computing *S′*, we start with *S* of *t* tracks and subsequently compute another feasible routing solution, *S**, of the same *t* tracks with reduced total crosstalk. To verify the performance of our algorithms, we start with the existing routing solutions, *S*, which are computed using *TAH* for several well-known benchmark examples of general channel specifications [16], [17]. The prime objective here is to compute *better* routing solutions in terms of total amount of crosstalk, obeying the constraints present in the example channels. In our first algorithm, we track-wise reassign the nets in a given routing solution, *S* so that a maximum amount of overall crosstalk is reduced in computing *S′* without violating the vertical constraints. In the second algorithm, nets that are re-assignable obeying both the constraints to some other track(s) are shifted so that the overall crosstalk is further reduced.

4.1 The First Algorithm

This algorithm is naturally evolved from the theory of reducing crosstalk, explained through Figures 1 and 2. If there is no vertical constraint in the given channel specification, then the algorithm for changing tracks designed in [15] and [26] is sufficient to reduce the overall crosstalk. Here, we first compute the effective spans of intervals of all the tracks in a given feasible d_{max}-track routing solution, S that is computed using any algorithm in [3], [18] for a simple channel instance. The effective span of intervals of track, i, is obtained by adding the actual spans of intervals of all the nets assigned to track, i in S. Then for a t-track routing solution, S we sort the tracks in descending order according to their effective spans of intervals. In this algorithm, we sandwich the

track, with the minimum effective span of intervals, within the tracks with the maximum and the next to maximum effective spans of intervals. Subsequently, we sandwich the track, with the next to minimum effective span of intervals within the tracks with the second and the third maximum effective spans of intervals, and so on. The flanked assignment of a track with less effective span of intervals by a pair of tracks with more effective spans of intervals is entirely motivated by the geometry of the channel and the initial routing solution provided as input to execute the algorithm. Note that any pair of nets that were assigned to the i-th track in S are now assigned to the j-th track in S' where i and j may or may not be the same. Since we have considered the general instances of channel specifications involving both the constraints, the situation is not as easy as stated above. Here we have to obey the presence of vertical constraints as well. As a consequence, we introduce the concept of reduced vertical constraint graph (RVCG) [16] as follows. We often represent vertical constraints by the RVCG that represents vertical constraints between groups of nets, where each group contains a set of non-overlapping intervals representing a clique in the HNCG. Since the nets in a track of a routing solution, S correspond to a clique in the HNCG, the RVCG has been used in this paper to represent vertical constraints between tracks in S. The formal definition of RVCG is as follows: each vertex in the vertex set corresponds to a set of non-overlapping nets. There is a directed edge (u, v), if there is a net $n_i \in u$ and another net, $n_j \in v$, such that (n_i, n_j) is a directed edge in the VCG. The algorithm starts with the RVCG of a given minimum area routing solution, S of t tracks of a channel, and in t iterative steps, track-wise reassigns the nets from top to bottom. Let RVC_i be the RVCG at the beginning of the i-th iteration, $1 \leq i \leq t$ and S_i be the set of source vertices in RVC_i. Note that before assignment of all the nets in computing S', S_i contains at least one element at the beginning of each iteration. This is because S is a feasible two-layer routing solution in the specified routing model without any cyclic vertical constraints, and the corresponding RVCG is also acyclic. In the i-th iterative step, $1 \leq i \leq t$, we select a source vertex, say s, in the current RVCG (i.e., RVC_i), so that the nets in s are best fitted (in terms of reduction in crosstalk) for their assignment to the i-th track from the top row of the channel. After assignment of all the nets in s to the i-th track, s and its adjacent edges are deleted from RVC_i and the RVCG, if not exhausted, for the next iteration is obtained. The algorithm is so self-reliant that it iterates exactly t times to compute S', for a given routing solution of t tracks. Now we state how each of the iterative steps works in computing S'. In the first iteration of our algorithm, we assign the nets corresponding to the source vertex, s such that the effective span of intervals of all the nets in s among the source vertices in RVC_1 is maximum. The idea of selecting this particular vertex, s for the topmost track is justified based on the assumption that the amount of crosstalk between the nets in the first track and the fixed terminals in the top row of the channel is negligible. Situations may demand the reverse, and the selection of s may vary accordingly. From the second track onwards in successive iterations, we select such a source vertex, s from RVC_i to assign the nets in s to the i-th track from the top of the channel, that renders minimum amount of crosstalk with the nets already assigned to the (i–1)-th track, $i \geq 2$. Now we state how a particular source vertex, s from RVC_i is selected to assign the corresponding nets present in it to the i-th track. In computing s, we apply a balanced binary search tree data structure among the set of source vertices in RVC_i in their effective spans of intervals. From this binary search tree we particularly trace two source vertices having the minimum and the maximum effective spans of intervals. This computation can easily be performed by identifying the end elements in the inorder sequence of the vertices in the search tree. According to this heuristic, the nets belonging to either the source vertex with the maximum effective span of intervals or the source vertex with the minimum effective span of intervals are best assignable to the i-th track. This can be computed in constant time by comparing crosstalks between the nets already assigned to the (i–1)-th track and the nets corresponding to the extreme source vertices in RVC_i. If these two effective spans intervals are same (this is true only when all the vertices in RVC_i are having the same effective span of intervals), we compute their total spans of intervals. The total span of intervals of the nets belonging to a track is the span between the starting column of the first net and the terminating column of the last net, i.e., the span of the track. This computation is motivated by considering the following aspects: (i) utilization of a track, (ii) congestion of nets over the region near the i-th track, (iii) long overlapping between the nets in adjacent tracks, and (iv) vertical wire length minimization. All these aspects are somehow incorporated in all the phases of the algorithms designed in this paper, and these are extremely important to synthesize VLSI physical design from performance driven routing point of view. Anyway, if the vertices are differentiated by their total spans of intervals, we select the one that is better fitted to the i-th track; otherwise we assign the nets of any one of them arbitrarily. This is clear from the algorithm stated above that the solution S' computed using this algorithm is a feasible routing solution of exactly t tracks, as it always assigns the nets from top to bottom of the channel and in each iteration of assigning the nets it selects a source vertex from the current RVCG. This completes the presentation of the first heuristic algorithm based on track interchange. Now we analyse the time complexity of this algorithm The RVCG from the given routing solution, S of t tracks is computed in $O(t + l) = O(n)$ time, where l is the length of the given channel specification and n is the number of nets in the channel, as both t and l are O(n). As the size of the set of source vertices in each iteration is O(n), the best fitted vertex from the balanced binary search tree in an iteration is computed in time O(logn). Modification of the RVCG takes O(n) time. Finally, as the algorithm iterates t times, the overall computational complexity of the algorithm is

O(nlogn) time for the channel specification with a total of n nets in the worst case.

4.2 The Second Algorithm

Note that the above algorithm we have designed is simple but efficient enough to reduce maximum amount of crosstalk belonging to a given routing solution of a channel, merely by reassigning the nets trackwise obeying vertical constraints. In the second algorithm, our objective is to interchange a pair of nets only when (i) the nets are horizontally constrained to each other, (ii) the interchange do not introduce any horizontal constraint violation due to overlapping with some other nets, (iii) the interchange do not introduce any vertical constraint violation in computing S', and (iv) the resulting crosstalk after interchanging the nets is reduced. This is not an easy task at all. Moreover, we do not know the sequence of interchanging pairs of nets so that a maximum amount of crosstalk is reduced. Furthermore, a particular net may be interchanged O(n) times among the tracks without giving any remarkable gain in overall crosstalk, and that might cause the problem of minimizing crosstalk drastically cost expensive. That is why in this algorithm without allowing net-to-net swapping, we shift a net to some other track where a suitable blank space is available and this shifting results in reduction of overall crosstalk. For some net, x if several such shifting is possible, we accept the shift of x that maximizes the reduction in crosstalk. In this heuristic, we sort the nets that are interchangeable from left to right in S with respect to their starting column positions in the channel. Then we consider the interchangeable nets one after another, and for some particular interchangeable net, x, we search out a track where the net is best fitted in terms of the overall crosstalk minimization without introducing any vertical constraint violation. Note that the sequence of interchanging the nets with a suitable blank space in some other track plays the most important role in reducing the maximum amount of crosstalk. In general, there is an exponential number of such sequences and it is not possible to consider all of them to maximize the reduction in crosstalk in computing S'. So we consider a constant number of such sequences to allow interchanging the position of a net with a blank space in some other track. Such sequences may be computed by sorting of the interchangeable nets (a) from top-left to bottom-right and (b) from bottom-left to top-right in S with respect to their starting column positions in the channel. Similarly, such sequences may be computed by sorting of interchangeable blank spaces (a) from top-left to bottom-right and (b) from bottom-left to top-right in S with respect to their starting column positions in the channel. Needless to mention that several such constant number of sequences may be considered, and allowed to go through this heuristic to accept the routing solution S' with the least amount of total crosstalk in it. Moreover, this heuristic may also be repeatedly followed a constant number of times if crosstalk is reduced in each case. This completes the presentation of the second heuristic algorithm. Next we analyse the time complexity of this algorithm. For a given t-track routing solution, S of n nets, this algorithm requires at most O(nt) time. Now since t = O(n), the algorithm takes $O(n^2)$ time in the worst case as for each of the O(n) interchangeable nets (or blank spaces) in S, the algorithm searches blank spaces (or nets) in at most O(n) tracks of the given routing solution, S of the channel.

5. Experimental Results and Discussion

Now we summarize the performance of our algorithms as follows. In this paper we have dealt with the general instances of channel specifications, and there are several such instances of benchmark channels including the famous Deutsch's Difficult Example (DDE). We have considered all these instances to quantify the amount of reduction in crosstalk in the computed routing solutions following the routing solutions obtained using TAH, the well-known Track_Assignment_Heuristic that is designed for computing minimum area routing solutions [16], [17]. We have considered all of the two-layer no-dogleg routing solutions computed using TAH, as given in Table 4.1 (Page # 102) and shown in different hardcopy routing solutions (Figures 4.4-4.17 in pages 103-112) of [16]. We have considered all these solutions as the initial routing solutions, and executed them through the algorithms developed in this paper one after another. In other words, we have computed the reduced crosstalk routing solutions for all the aforesaid benchmark examples following a two-phase implementation, wherein the first phase solutions are computed through the first algorithm. Then these solutions are subsequently inputted to compute further reduced crosstalk routing solutions following the second algorithm. All the results are included in Table 1.

Table 1: Amount of crosstalk computed after each of the algorithms and percentage reduction in overall crosstalk.

Example	*Number of Tracks*	*Amount of Crosstalk*			
		TAH	*First Algorithm*	*Second Algorithm*	*%age Reduction*
Ex. 1	12	201	201 (0)	196 (1)	02.49
Ex. 2	15	414	397 (1)	396 (1)	04.35
Ex. 3(a)	16	564	519 (1)	506 (1)	10.28
Ex. 3(b)	18	602	507 (1)	502 (1)	16.61
Ex. 3(c)	19	795	771 (1)	732 (3)	07.92
Ex. 4(b)	19	953	901 (1)	845 (2)	11.33
Ex. 5	20	942	724 (1)	675 (2)	28.34
DDE	29	1510	1435 (1)	1181 (3)	21.79
r1	23	1519	1482 (1)	1421 (2)	06.45
r2	20	1071	1034 (1)	1034 (0)	03.45
r3	18	784	728 (1)	690 (3)	11.99
r4	18	1262	1260 (1)	1206 (1)	04.44
Ex. 3(b).1	21	518	504 (1)	393 (2)	24.13
Ex. 3(c).1	18	846	818 (1)	775 (3)	08.39

Amount of crosstalk following algorithm TAH is the initial amount of crosstalk [16] as shown in column TAH. Each of the columns First Algorithm and Second Algorithm shows the computed amount of crosstalk following the corresponding algorithms. The first algorithm computes a drastically reduced crosstalk routing solutions from the initial routing solutions computed using TAH as the first phase of implementation, and the second algorithm computes the mostly reduced crosstalk routing solutions based on the routing solutions computed using the first algorithm. Digits within parentheses in these columns indicate the number of times further the related algorithm has been executed in obtaining a better routing solution in terms of total amount of crosstalk. Note that r3 is the only example that has been executed twice more following the first algorithm; all the remaining examples are executed at most once more to obtain the best possible solution following this algorithm.

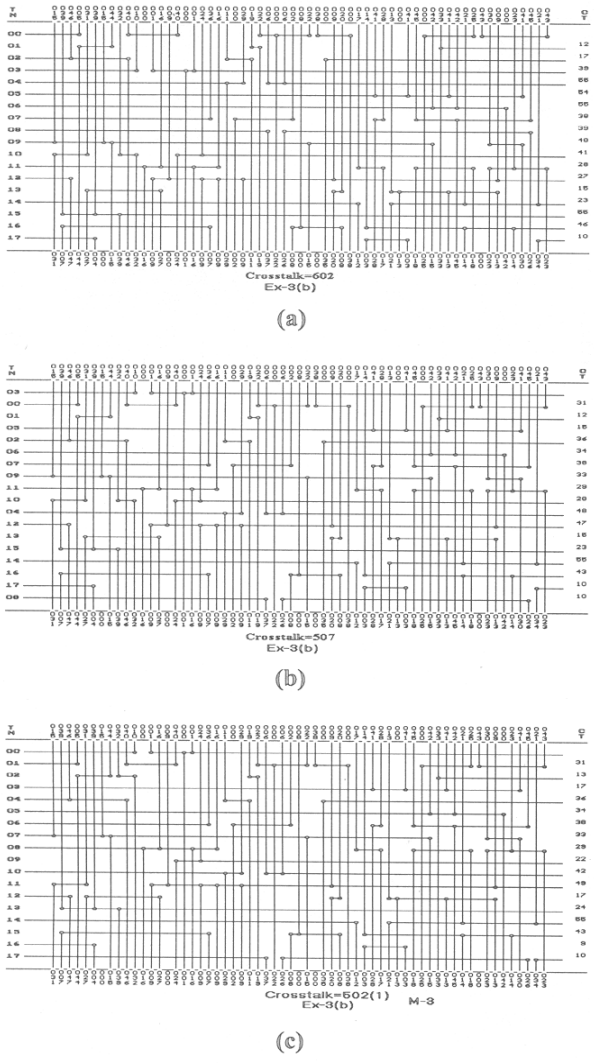

Figure 3 (a) A minimum area routing solution for Ex. 3(b) using algorithm TAH. (b) A minimum crosstalk routing solution for Ex. 3(b) using the first algorithm. (c) A further minimum crosstalk routing solution for Ex. 3(b) using the second algorithm.

Furthermore, this algorithm fails to improve the solution for example Ex. 1, as the initial solution after TAH is the desired solution after the implementation of the first algorithm. Then the second algorithm is executed successively for an example until it computes better routing solutions in terms of reduction in crosstalk. This succession is the maximum of three times more for examples Ex. 3(c), DDE, r3, and Ex. 3(c).1, and incidentally there is no scope of shifting nets for example r2 after the solution computed following the first algorithm. Percentage reduction (in crosstalk) column is obtained by computing the overall reduction in crosstalk, starting from the initial routing solution computed using TAH [16]. Note that the amount of crosstalk reduction in a routing solution exclusively depends upon the factors of (i) congestion of nets and constraints involving with the nets in the example channel and (ii) the initial routing solution under consideration. In all these respects the routing solutions we have obtained following the algorithms developed in this paper are highly encouraging. For example channel Ex. 5, the reduction in overall crosstalk is 28.34%, which is the maximum among the example channels under consideration. For the famous DDE, the overall reduction in crosstalk is 21.79%, which is very much interesting and inspiring from the point of view of high performance routing in VLSI physical synthesis. The hardcopy routing solutions for Ex. 3(b), Ex. 5, DDE, r3, and Ex. 3(b).1 are shown in Figures 3-7, respectively. Each of the figures contains three routing solutions: (a) the initial routing solution that was computed using TAH [16], [17], (b) the routing solution computed using the first algorithm, and (c) the routing solution computed using the second algorithm. Hardcopy solution (b) in each of the figures shows the reassignment of tracks indicating the initial track numbers (TN) at the left of the same. Moreover, in each of the hardcopy solutions, the amount of crosstalk (CT) between the nets assigned to adjacent tracks is shown at the right of the solution.

6. CONCLUSION

In this paper we have considered the problem of crosstalk minimization in two-layer channel routing, which is an NP-hard problem for general instances of channel specifications. We have designed two heuristic algorithms to minimize crosstalk starting from a given routing solution. The first algorithm runs in time O(nlogn) whereas the second algorithm runs in O(n2) time, where n is the number of nets belonging to the channel. Here we like to point out a few possible extensions of our work and/or open problems as follows.

(i) In the second algorithm, we have reassigned a net into a blank space in some other track, if reduction in crosstalk is achieved in doing so. A generalised version of this algorithm may produce better routing solution(s) in terms of crosstalk minimization when, satisfying constraints, nets are free to interchange among the tracks.

(ii) Instead of computing the total crosstalk one might be interested in computing every local crosstalk between a pair of nets such that no local crosstalk would cross a

given limit (i.e., number of units of overlapping between a pair of intervals of the nets assigned to two adjacent tracks). This is interesting as a problem of combinatorial optimization, and equally important from high performance design of routing point of view.

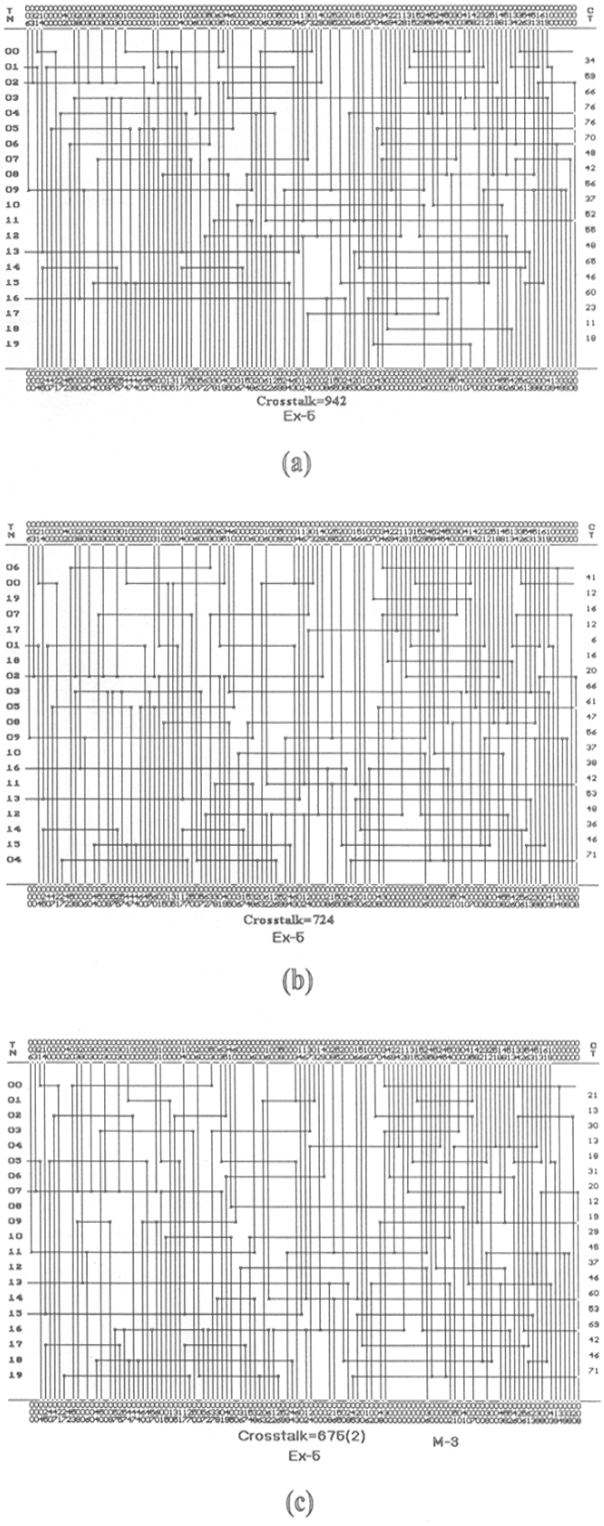

Figure 4 (a) A minimum area routing solution for *Ex. 5* using algorithm *TAH*. **(b)** A minimum crosstalk routing solution for *Ex. 5* using the first algorithm. **(c)** A further minimum crosstalk routing solution for *Ex. 5* using the second algorithm

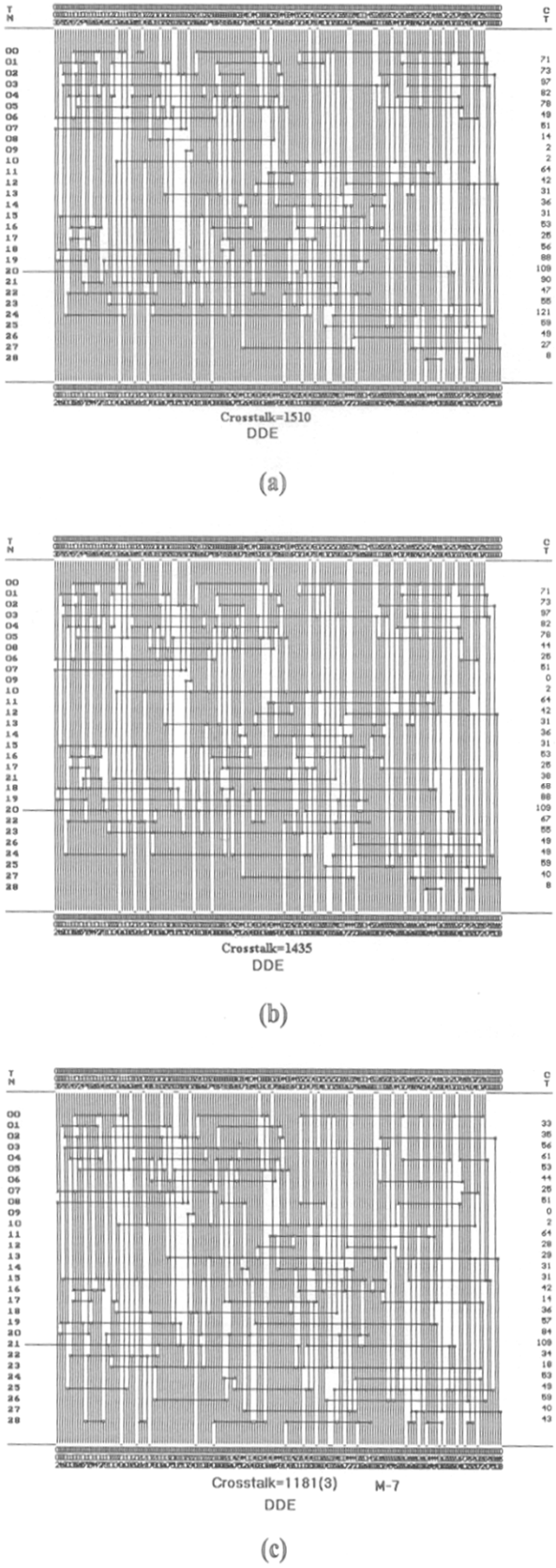

Figure 5 (a) A minimum area routing solution for *DDE* using algorithm *TAH*. **(b)** A minimum crosstalk routing solution for *DDE* using the first algorithm. **(c)** A further minimum crosstalk routing solution for *DDE* using the second algorithm.

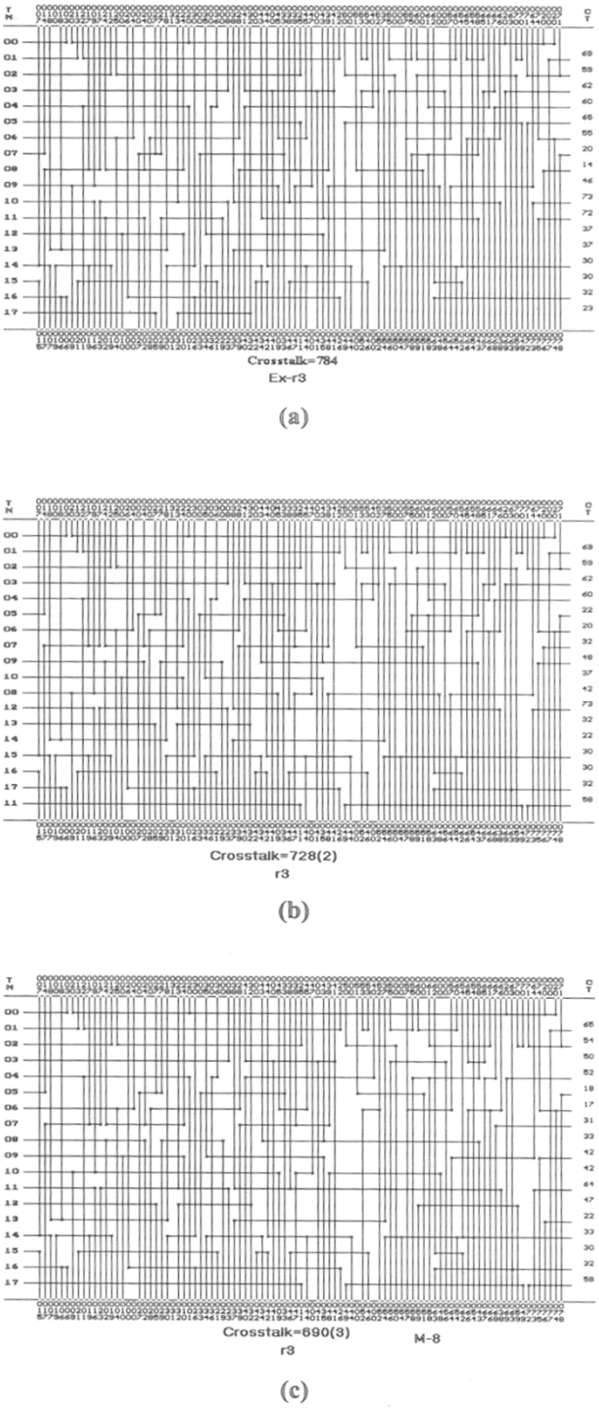

Figure 6 (a) A minimum area routing solution for *r3* using algorithm *TAH*. **(b)** A minimum crosstalk routing solution for *r3* using the first algorithm. **(c)** A further minimum crosstalk routing solution for *r3* using the second algorithm.

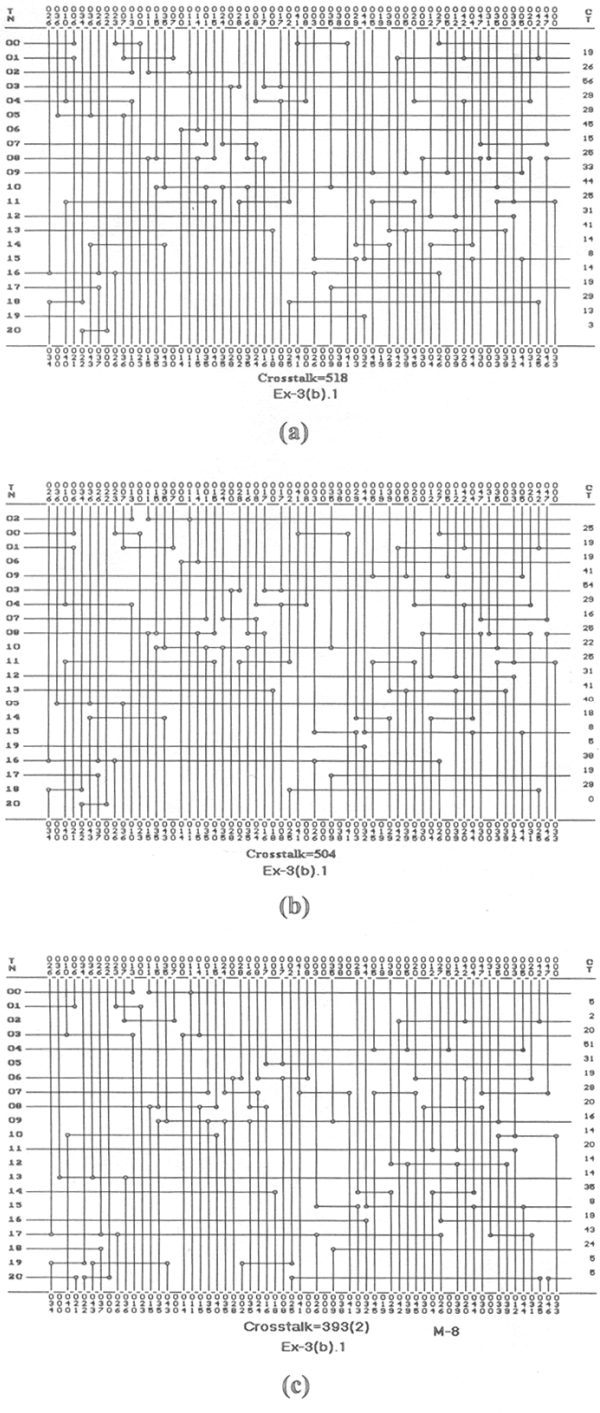

Figure 7 (a) A minimum area routing solution for Ex. 3(b).1 using algorithm TAH. **(b)** A minimum crosstalk routing solution for Ex. 3(b).1 using the first algorithm. **(c)** A further minimum crosstalk routing solution for Ex. 3(b).1 using the second algorithm.

(iii) Researchers may also be interested in computing minimized crosstalk routing solutions in expense of permissible more channel area. In any case, this area is no way double of the minimum (or optimal) area required.

(iv) Instead of starting from a given routing solution researchers may also compute good routing solutions directly optimizing crosstalk and some other cost optimization objective of CRP.

(v) Minimized crosstalk routing solutions in the cases of three- and multi-layer channel routing might draw the current interest of research. Also doglegging may also be introduced in all these cases.

References

[1] Ernesto Rayas-Sánchez, J. "A frequency-domain approach to interconnect crosstalk simulation and minimization." Microelectronics Reliability 44, no. 4 (2004): 673-681.

[2] Gao, T. and C. L. Liu. "Minimum crosstalk channel routing." IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, no. 5 (1996): 465-474.

[3] Hashimoto, A. and J. Stevens. "Wire routing by optimizing channel assignment within large apertures." Proc. of the 8th Design Automation Workshop, pp. 155-169. ACM, 1971.

[4] Ho T.-T., S. S. Iyengar, and S.-Q. Zheng. "A general greedy channel routing algorithm." IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 10, pp. 204-211, 1991.

[5] LaPaugh A. S. "Algorithms for Integrated Circuit Layout: An Analytic Approach." Ph.D. Thesis, Lab. for Computer Sc., MIT, Cambridge, 1980.

[6] Liberali, V., R. Rossi, and G. Torelli. "Crosstalk effects in mixed-signal ICs in deep submicron digital CMOS technology." Microelectronics Journal, vol. 31, no. 11 (2000): 893-904.

[7] Moghaddam, Soodeh Aghli, and Nasser Masoumi. "Analysis and simulation of a novel gradually low-K dielectric structure for crosstalk reduction in VLSI." Microelectronics Journal 39, no. 12 (2008): 1751-1760.

[8] Maheswari, M., and G. Seetharaman. "Multi-bit random and burst error correction code with crosstalk avoidance for reliable on chip interconnection links." Microprocessors and Microsystems, vol. 37, no. 4 (2013): 420-429.

[9] Manem, H., and G. S. Rose. "A crosstalk minimization technique for sub-lithographic programmable logic arrays." Proc. of 9th IEEE Conference on Nanotechnology, pp. 218-221, 2009.

[10] Mitra, P., N. Ghoshal, and R. K. Pal. "A graph theoretic approach to minimize total wire length in channel routing." Proc. of 18th IEEE Region 10 International Conference on Convergent Technologies for the Asia-Pacific (IEEE TENCON 2003), vol. 1, pp. 414-418, 2003.

[11] Moiseev, K., S. Wimer, and A. Kolodny. "Timing-constrained power minimization in VLSI circuits by simultaneous multilayer wire spacing." Integration, the VLSI Journal (2014).

[12] Pal, A., A. K. Khan, S. Saha Sau, A. K. Dutta, R. K. Pal, and A. Chaudhuri. "Application of graph in computing reduced area VLSI channel routing solutions." Proc. of International Conference on Computing and Systems (ICCS-2010), pp. 249-256. 2010.

[13] Pal, A., T. N. Mandal, A. Khan, R. K. Pal, A. K. Datta, and A. Chaudhuri. "Complexity issues and design of algorithms for crosstalk minimization in two-layer channel routing." manuscript. 2014.

[14] Pal, A., T. N. Mandal, S. Saha Sau, A. K. Datta, R. K. Pal, and A. Chaudhuri. "Graphs – The tool to visualize the problems in VLSI channel routing." Assam University Journal of Science and Technology, vol. 7, no. 2 (2011): 73-83.

[15] Pal, A., A. Singha, S. Ghosh, and R. K. Pal. "Crosstalk minimization in two-layer channel routing." Proc. of 17th IEEE Region 10 International Conference on Computers, Communications, Control and Power Engineering (IEEE TENCON 2002), vol. 1, pp. 408-411, 2002.

[16] Pal R. K. "Multi-Layer Channel Routing: Complexity and Algorithms." Narosa Publishing House, New Delhi (Also published from CRC Press, Boca Raton, USA and Alpha Science International Ltd., UK), 2000.

[17] Pal, R. K., A. K. Datta, S. P. Pal, M. M. Das, and A. Pal. "A general graph theoretic framework for multi-layer channel routing." Proc. of the IEEE 8th International Conference on VLSI Design, pp. 202-207, 1995.

[18] Pal, R. K., A. K. Datta, S. P. Pal, and A. Pal. "Resolving horizontal constraints and minimizing net wire length for multi-layer channel routing." Proc. of IEEE Region 10's Eighth Annual International Conference on Computer, Communication, Control and Engineering (TENCON'93), vol. 1, pp. 569-573, 1993.

[19] Pal, R. K., and A. Pal. "An efficient graph-theoretic algorithm for three-layer channel routing." Proc. of the IEEE 5th International Conference on VLSI Design, pp. 259-262, 1992.

[20] Pal, R. K., S. P. Pal, A. Pal, and A. K. Dutta. "NP-completeness of multi-layer no-dogleg channel routing and an efficient heuristic." Proc. of the IEEE 6th International Conference on VLSI Design, pp. 80-83, 1993.

[21] Pande, P. P., A. Ganguly, H. Zhu, and C. Grecu. "Energy reduction through crosstalk avoidance coding in networks on chip." Journal of Systems Architecture, vol. 54, no. 3 (2008): 441-451.

[22] Reis, J. D., M. V. Drummond, A. L. Teixeira, R. N. Nogueira, P. Monteiro, S. Shinada, N. Wada, and G. M. Beleffi. "Experimental demonstration of a nonlinear effects crosstalk minimization algorithm." Proc. of National Fiber Optic Engineers Conference, p. JThA32. Optical Society of America, 2010.

[23] Sankaran, H., and S. Katkoori. "Simultaneous scheduling, allocation, binding, re-ordering, and encoding for crosstalk pattern minimization during high-level synthesis." IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 2 (2011): 217-226.

[24] Sathish, A., and M. Madhavi Latha. "Data encoding technique for crosstalk delay reduction on fault tolerant data-bus in DSM technology." Procedia Engineering, vol. 38 (2012): 2967-2972.

[25] Schaper G. A. "Multi Laver Channel Routing." Ph.D. Thesis, Dept. of Computer Sc., Univ. of Central Florida, Orlando, 1989.

[26] Singha A., S. Ghosh, A. Pal, and R. K. Pal. "High performance routing for VLSI circuit synthesis." Proc. of 6th IEEE VLSI Design and Test Workshops 2002 (IEEE VDAT2002), pp. 348-351, 2002.

[27] Szymanski, T. G. "Dogleg channel routing is NP-complete." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 4, no. 1 (1985): 31-41.

[28] Terapasirdsin, A., and N. Wattanapongsakorn. "Crosstalk minimization in VLSI design using signal transition avoidance." Proc. of the International Symposium on Communications and Information Technologies (ISCIT), pp. 911-915, 2010.

[29] Verma, S. K., and B. K. Kaushik. "Crosstalk and power reduction using bus encoding in RC coupled VLSI interconnects." Proc. of the 3rd IEEE International Conference on Emerging Trends in Engineering and Technology (ICETET), pp. 735-740, 2010.

[30] Xu, J., X. Hong, T. Jing, L. Zhang, and J. Gu. "A coupling and crosstalk-considered timing-driven global routing algorithm for high-performance circuit design." Integration, the VLSI Journal, vol. 39, no. 4 (2006): 457-473.

[31] Yoshimura, T., and E. S. Kuh. "Efficient Algorithms for Channel Routing." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 1, no. 1 (1982): 25-35.

AUTHORS

Achira Pal received the B.Sc. and M.Sc. degrees in Physics in 1986 and 1988, respectively and the M.Tech. degree in Atmospheric Sciences in 1994 from the University of Calcutta. She is currently pursuing her Ph.D. in the Department of Computer Science and Engineering, Jadavpur University. Her research interests include VLSI physical design and Algorithms.

Tarak Nath Mandal received the B.Sc. degree in Computer Science and the B.Tech. degree in Computer Science and Engineering from the University of Calcutta in 2003 and 2006, respectively. He is currently pursuing his Ph.D. in the Department of Computer Science and Engineering, University of Calcutta. Since 2006, he has been with IBM India Pvt. Ltd. in various roles: Team Lead / Coordinator, SAP ABAP Consultant, Solution Architect, etc.

Abhinandan Khan received his B.Tech. degree in Electronics and Communication Engineering from the West Bengal University of Technology in 2011 and the M.E. degree in Electronics and Telecommunication Engineering from Jadavpur University in 2013 with specialisation in Electron Devices. He is currently pursuing his Ph.D. in the Department of Computer Science and Engineering, University of Calcutta. His research interests include Soft computing, Bioinformatics, Artificial intelligence, and so on.

Rajat Kumar Pal received the B.E. degree in Electrical Engineering in 1985 and the M.Tech. degree in Computer Science and Engineering in 1988 from the University of Calcutta, and the Ph.D. degree in Computer Science and Engineering in 1996 from the Indian Institute of Technology, Kharagpur. Since 1994, he has been as a faculty with the Department of Computer Science and Engineering, University of Calcutta. He is currently a Professor with the Department of Computer Science and Engineering, University of Calcutta. His research interests include VLSI design, Graph theory, Perfect graphs, Logic synthesis, Design of algorithms, and so on.

Alak Kumar Datta received the B.Sc. and M.Sc. degrees in Mathematics from the University of Calcutta, in 1985, and the Indian Institute of Technology, Kharagpur, in 1987, respectively. He received his Ph.D. from the Indian Institute of Technology, Kharagpur in 1995. From 1995 to 2007, he was a faculty at the Bengal Engineering and Science University, Shibpur. At present, he is a Professor in the Department of Computer and System Sciences, Visva-Bharati. His research interests include Parallel algorithms, Graph algorithms, Approximation algorithms, VLSI channel routing algorithms.

Atal Chaudhuri received the B.E. and M.E. degrees in Electronics and Telecommunication Engineering and the Ph.D. degree from Jadavpur University in 1980, 1982, and 1989, respectively. During 1985-1988, he was a Lecturer in the Department of Electronics and Telecommunication Engineering, Jadavpur University. He is currently a Professor with the Department of Computer Science and Engineering, Jadavpur University. His current research interests are Mobile ad-hoc network and Information security