Journal of Physical Sciences, Vol. 10, 2006, 167 – 177

Algorithms for Reducing Crosstalk in Two-Layer Channel Routing

Achira Pal*[+]*, Debojit Kundu*[*]*, Alak K. Datta*[#]*,
***Tarak N. Mandal*[*] *and Rajat K. Pal*[*]**
[+] Harinavi Subhasini Balika Sikshalaya, PO: Harinavi, Dist.: 24 Parganas (South), PIN: 743 359, West Bengal, India
[#] Department of Computer Science and Technology, BESU, PO: Botanic Garden, Howrah – 711 103, West Bengal, India
[*] Department of Computer Science and Engineering, University of Calcutta, 92, A. P. C. Road, Kolkata – 700 009, India

Received 15 September, 2006 ; accepted 16 December, 2006

ABSTRACT

Crosstalk minimization is one of the most important high performance aspects in interconnecting VLSI circuits. With advancement of fabrication technology, devices and interconnecting wires are placed in closer proximity and circuits operate at higher frequencies. This results in crosstalk between wire segments. Crosstalk minimization problem for the reserved two-layer Manhattan channel routing is NP-hard, even if the channel instances are free from any vertical constraint (simplest channel instances). In this paper we have developed heuristic algorithms for computing reduced crosstalk two-layer channel routing solutions for simplest as well as general channel instances. In general, the results obtained are highly encouraging.

Keywords : Channel routing problem, Crosstalk minimization, NP-hardness, *Simplest* channel instance, High performance routing, Heuristic algorithm.

1. Introduction

1.1. Foundation of the problem

In VLSI layout design, it is required to realize a specified interconnection of a set of terminals present in different modules, primarily using minimum possible area. This is known as *routing problem*. There exist several routing strategies for efficient interconnections among different modules. One of the most important types of routing strategies is *channel routing* [1, 3, 4, 7, 10, 11].

A *channel* is a rectangular routing region that has two open ends (left and right) and has two rows (upper and lower) of fixed points called *terminals*. Terminals are assumed to be aligned vertically in columns that are usually equispaced along the length of the channel. A set of terminals that need to be electrically connected together is called a *net*. A vertical wire segment is a wire that lies in a *column*, whereas a horizontal wire segments is a wire that lies in a *track* in a grid-based reserved layer Manhattan routing model [4, 7, 11]. Tracks are horizontal lines that are usually equispaced along the height of the channel, parallel to the two rows of (fixed) terminals.

A *route* for a net is a collection of horizontal and vertical wire segments spread across the different layers connecting all the terminals of the net. A *legal wiring* of a channel is a set of routes that satisfy all the prespecified conditions where, no two wire segments used to connect different nets overlap on the same conducting layer. A legal wiring is also called a *feasible routing solution*.

The *Channel Routing Problem (CRP)* is specified by two *m*-element vectors *TOP* and *BOTTOM*, and a number *t*; objective is to find a feasible routing solution for the channel using no more than *t* tracks, if it exits [7]. Let L_i (R_i) be the leftmost (rightmost) column position of net n_i, then $I_i=(L_i,R_i)$ is known as its interval (or span).

1.2. Crosstalk and High Performance Routing

As fabrication technology advances and feature size reduces, devices are placed in closer to each other, and interconnecting wire segments are assigned with narrower pitch, whereas circuits' operations are realized at higher frequencies. As a result, electrical hazards viz., *crosstalk* between wire segments is evolved. Crosstalk between wire segments is proportional to the coupling capacitance, which is in turn proportional to the *coupling length*; the total length of overlap between wire segments of two different nets on adjacent tracks. Crosstalk is also proportional to the frequency of operation and inversely proportional to separating distance between wires.

More crosstalk means more signal delay and reduced circuit performance. Therefore, in high performance routing it is desirable to develop channel routing algorithms that not only compute minimum area channel routing solutions but also reduce crosstalk. We define the amount of crosstalk between horizontal wire segments of two different nets assigned to two adjacent tracks in a given routing solution is proportional to the amount of overlap of their horizontal spans. If two intervals do not overlap, there is no horizontal constraint between the nets. That is, if there is an overlap of the horizontal wire segments of a pair of nets, there is a possibility of having crosstalk between them.

We measure crosstalk in terms of number of units of overlap between a pair of nets that are assigned to adjacent tracks in a feasible routing solution. We assume that the crosstalk between wire segments of two different nets assigned to two nonadjacent tracks is negligibly small, and hence can be ignored. We also assume that the amount of crosstalk between vertical wire segments of two different nets placed in two adjacent columns to be very small, and hence can be neglected (as usually the number of columns in a channel is much more than the number of tracks required in it).

1.3 Some Definitions

Now we define a few terms and constraints involving CRP, and characterize the problem as stated below. In this paper, we allow all interconnections of nets present in a channel using two-layer Manhattan routing model [4, 7, 11], where one layer is reserved for horizontal wire segment only and the other is reserved for vertical wire segment only. Such a routing model is known as the two-layer *VH* routing model. In this model, the CRP is characterized by two important constraints, viz., the *horizontal constraint* and the *vertical constraint* [7, 11]. These constraints are usually represented by two graphs, viz., the *horizontal constraint graph (HCG)* and the *vertical constraint graph (VCG)*, respectively [7, 11]. The HCG is simple undirected graph. Specifically, the HCG is an interval graph and, therefore, a perfect graph [2]. The VCG is an arbitrary directed graph that may or may not contain any cycle. *Reduced vertical constraint graph (RVCG)* is another directed graph that can be formed from *VCG* [8]. The *local density* of a column (in a channel) is the maximum number of nets passing through it. The *channel density* is the maximum of all the local densities in a channel [7, 11]. We denote channel density by d_{max}.

2. Crosstalk Minimization Problem

In CRP, usually the prime intention of a router is to compute such a solution that uses a minimum number of tracks (or minimum channel area). In addition, in high performance routing our interest is also to obtain a routing solution with less electrical hazards (i.e., crosstalk), less signal propagation delay, less power consumption, less or no hotspot formation, etc.

Crosstalk is one of the most important high performance optimization criteria in channel routing that should be reduced to get better performance in routing. In this paper we have developed algorithms for minimizing crosstalk for reserved two-layer (VH) Manhattan channel routing model [7, 11]. There are two types of crosstalk minimization problem, namely *sum-crosstalk minimization* and *bottleneck-crosstalk minimization* [6]. Sum-crosstalk is the amount of total crosstalk between horizontal wire segments of the nets that are assigned to adjacent tracks. The sum-crosstalk minimization problem is to compute a feasible routing solution with a given number of tracks in which the total amount of crosstalk is minimized. Similarly, bottleneck-crosstalk with respect to a feasible routing solution is the maximum amount of crosstalk due to overlapping between any pair of adjacent horizontal wire segments of two different nets. So the bottleneck crosstalk minimization problem of finding a feasible routing solution with a given number of tracks, such that the bottleneck-crosstalk is minimized. Here in this paper we have developed algorithms only for sum-crosstalk minimization problem.

The CRP of area minimization being an *NP*-hard problem [5, 10]; several heuristics have been proposed for routing channels in different routing models [4, 7, 9, 11]. The problem is polynomial time solvable if the channel instances are free from any vertical constraint, and there are algorithms for computing d_{max}-track routing solutions for such instances [3, 8]. Since the problem of minimizing area for an instance of routing channel with only horizontal constraints is polynomial time solvable (using d_{max} tracks), we define such instances as *simplest channel instances* of channel routing. We define a channel specification as *general*, if both the constraints are present in it.

However, the crosstalk minimization problem for two-layer routing, both in case of *simplest* as well as *general* channel instances are *NP*-hard [6], i.e., there exists no polynomial time algorithm for crosstalk minimization in two-layer channel using the routing model under consideration. In this paper we have developed heuristic algorithms for two-layer channel routing for simplest as well as general instances of channel specifications, as stated in the following section.

3. Algorithms for Crosstalk Minimization

In the previous section we have mentioned that crosstalk minimization problem for two-layer channel routing is *NP*-hard, even if the channel instances are *simplest*. So, first we develop crosstalk minimization algorithm for two-layer channel routing, where instances are free from any vertical constraint. Then we extend it for two-layer routing with *general* channel instances. Before that let us consider a simplest channel instance of only three nets and illustrate the presence of crosstalk between nets (or intervals), when these are assigned to tracks in a two-layer VH routing model (see Figure 1). Interestingly, the fact to be noticed that just by reassigning the nets to tracks, the amount of crosstalk in Figure 1(b) is reduced to 30.77% to that of in Figure 1(a). Hence we have the following observation.

Observation 1: *The amount of crosstalk is mostly reduced if a net (or interval) of smaller span is sandwiched by two nets (or intervals) of larger spans, or vice versa.*

This observation is the motivation in developing the first algorithm.

3.1. Algorithm *track interchange* (for *simplest channel instances*)

First of all we compute a d_{max}-track feasible routing solution either using *LEA*, or *MCC*1, or *MCC*2 [3, 8]. Then we reassign the nets trackwise so that the total amount of crosstalk is reasonably reduced. In order to do so, we define the term *effective interval* of nets that are assigned to a track. The *effective interval (EI)* of a track (for the nets assigned to it) is computed by adding the net spans of all the nets belonging to the track. Let us assume that we have an instance of the CRP with at most *t* nets overlapping each other. We sort the tracks according to their *EI* in *descending* order. It may happen that two or more tracks having the same *effective intervals*. In such cases, we sort these *equal EI* tracks based on their *total intervals (TI)* (i.e., the interval between the leftmost terminal and the rightmost terminal of the net(s) assigned to a track) in *ascending* order. It is done so, because, in between two tracks with same *EI*, the one with larger *TI* is sparser than that of the other track; so the track with more *TI* should come later in the computed sorted order (using *EI*). To sort the tracks based on the criteria with *primary key*: *EI* and *secondary key*: *TI*, we may use any sorting algorithm that runs in O(n^2) or lesser time. Now we trackwise rearrange the nets in a sandwiched fashion as stated below. For a set of t sorted tracks $\{n_1, n_2, \ldots, n_t\}$, the rearranged sequence is $\{n_1, n_t, n_2, n_{t-1}, n_3, n_{t-2}, \ldots\}$. A more improvised rearranging sequence is $\{n_1, n_t, n_3, n_{t-2}, \ldots, n_{t-3}, n_4, n_{t-1}, n_2\}$. Reduction of crosstalk in both these cases is more or less same; however, the later sequence is more desirable in practice. In this paper, though we are interested only with crosstalk between nets assigned to adjacent tracks, but as a high performance optimization criterion, we should reduce crosstalk that occurs between non-adjacent tracks also; the later rearranging sequence optimizes that too.

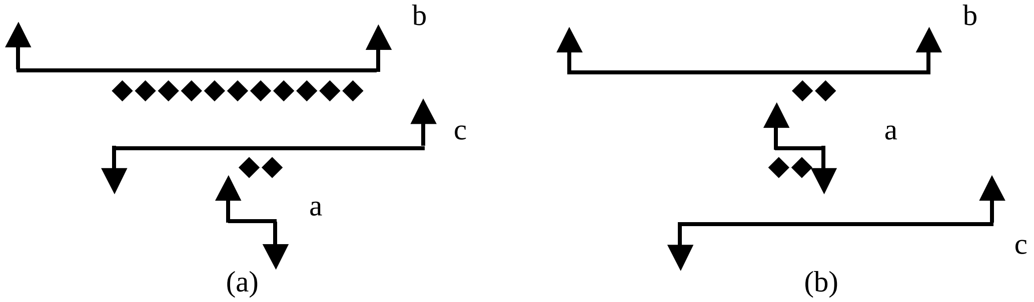


Figure 1: Crosstalk minimization in two-layer VH channel routing, in the absence of vertical constraints. **(a)** A feasible three-track routing solution with three intervals of three different nets a, b, and c that are overlapping to each other. Nets b and c share 11 units of horizontal span in the 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.

3.1. Algorithm *track interchange* (for *general channel instances*)

Here channel instances contain vertical constraints that are represented by VCG, which is a directed graph $VC=(V,A)$, where a net n_i is represented by a vertex $v_i \in V$, and a directed edge $(v_i,v_j) \in A$ (or directed path from vertex v_i to vertex v_j in VC) represents that the net n_i is to be assigned to a track above the track to which the net n_j is assigned. In order to route the nets and obtain feasible routing solutions, the vertical constraints should be maintained.

Here we cannot apply the crosstalk minimization algorithm (as in Section 3.1) directly, because, free rearranging the tracks may introduce undesired vertical constraint violation in the case of general channel instances. So we need to compute a group of *freely interchangeable* tracks (that are not vertically constrained to each other) for a given routing solution on which we can apply our algorithm *Track Interchange* (see Section 3.1) in one iteration. This process of computing freely interchangeable tracks and their assignment to tracks in successive iterations is continued until all the nets are assigned, and may result in obtaining a reduced crosstalk routing solution of the given general channel instance.

A set $T=\{t_1, t_2, \ldots, t_k\}$ of t tracks is called *freely interchangeable,* if there is no vertical constraints between any pair of nets (n_i,n_j), where nets n_i and n_j are assigned to two different tracks t_p and t_q, respectively, such that t_p, $t_q \in T$. In order to get a group of such freely interchangeable tracks, we can use a modified VCG, called RVCG, where each vertex represents a set of nets that are assigned to a track. So, the number of vertices in RVCG is same as the number of tracks needed to route the channel. If nets n_i and n_j are assigned to tracks t_p and t_q, and there is an edge (or a path) from n_i to n_j (i.e., n_j is successor of n_i) in *VC*, then there should be an edge from t_p to t_q in the RVCG. As an RVCG is computed from a given two-layer feasible routing solution, it is a directed acyclic graph (DAG) without any parallel edge between any pair of vertices. We use the terms, an *RVCG vertex* and a *track* interchangeably.

So, from the computed RVCG, we first identify the set T_1 of source vertices (that are freely interchangeable), and assign them to the topmost $|T_1|$ tracks using algorithm in Section 3.1. We delete the corresponding vertices from the RVCG, and compute another set T_2 of source vertices (that are freely interchangeable). We apply the same algorithm (in Section 3.1) and assign them to the next $|T_2|$ tracks of the channel. This process is continued until trackwise all the nets are reassigned to tracks.

A modified version of this algorithm is greedy in nature. Here instead of considering the set of source vertices at a time (in one iteration) for their assignment to tracks, we select the source vertex in the RVCG having maximum *EI* among all the sources in it. We assign it to the topmost track, and modify the RVCG by deleting the selected vertex. Then in the next iteration, we select the source vertex from the modified RVCG whose assignment renders minimum crosstalk, and delete the vertex from the modified RVCG in order to start the third iteration. This process is continued until all the vertices of the RVCG are considered for their assignment to tracks in subsequent iterations.

3.3 Algorithm *Net-Change*

Algorithm *Net-Change* is basically a greedy algorithm where each net after their assignment following algorithm *Track Interchange* is considered one after another in some sequence. Then it is checked whether net n_i with span I_i assigned to track t_p is reassignable to some other track (other than t_p) so that all the constraints are satisfied but crosstalk is reduced. This algorithm may result in reducing crosstalk further. In practice, we successively call algorithms *Track Interchange* and *Net-Change* in order to compute a mostly reduced crosstalk routing solution for general channel instances. The computational complexity of this algorithm is $O(n^2)$, where n is the number of nets belonging to the channel.

4. Experimental Results

As *simplest* as well as *general* benchmark channel instances are not sufficient to execute the algorithms developed in this paper, we create a large number of random channel instances of both types. In order to make the generated instances random in nature along the length of the channel, we follow some criteria as stated below.

(a) Though it is needless to mention that the net numbers are nothing but symbols to differentiate themselves, in our randomly generated channel instances the net numbers would also present randomly, i.e., the nets are not sorted in succession based on their starting column positions from left to right, or something of that sort.

(b) Nets are having different spans (or intervals) and they are also present randomly. This criterion tells us that all the smaller (or larger) nets are not accumulated (or concentrated) on a side of the channel.

(c) Generally, in practice, a channel contains a large number of smaller nets and less larger nets. Here the smaller or larger nets are differentiated based on their relative spans (or intervals) in a channel. Obviously, the number of nets with some intermediate spans is neither more nor less. Actually, this criterion helps us in generating random channel instances where the number of nets gradually reduces as their spans increase along the length of the channel.

Results computed are shown in Tables 1 and 2; Table 1 contains the results of simplest channel specifications whereas Table 2 shows the results of general channel specifications. For each case of number of nets we generate 200 random instances, and the data in a row are obtained by making average of each set of 200 executed data.

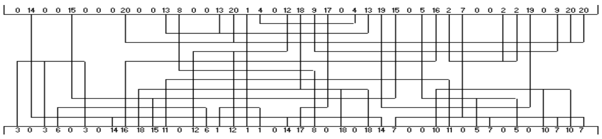
Two sets of routing solutions for general channel instances are shown in Figures 2 and 3. In each figure, (a) gives the initial amount of crosstalk using some algorithm, (b) gives the amount of crosstalk after algorithm *Track Interchange*, and (c) gives the amount of crosstalk after algorithm *Net Change*. The number of nets in Figure 2 (Figure 3) is 20 (40), whereas the length of the randomly generated channel is 43 (89).

Number of Nets	**Initial Crosstalk after *MCC*1**	**Crosstalk after *Reassignment to Tracks***	**Reduction after *Reassignment to Tracks* (%)**	**Crosstalk after *Net Change* (%)**	**Reduction after *Net Change* (%)**
10	25	16	36.00	16	36.00
15	69	47	31.88	46	33.33
20	131	90	31.30	88	32.82
25	221	155	29.86	153	30.77
30	322	226	29.81	223	30.75
35	465	330	29.03	326	29.89
40	611	439	28.15	434	28.97
45	797	569	28.61	562	29.49
50	1012	733	27.57	726	28.26
60	1483	1089	26.57	1079	27.24
70	2045	1522	25.57	1509	26.21
80	2745	2031	26.01	2016	26.56
90	3484	2618	24.86	2598	25.43
100	4290	3196	25.50	3173	26.04
110	5264	3945	25.06	3919	25.55
120	6284	4784	23.87	4754	24.35
130	7450	5590	24.97	5557	25.41
140	8661	6588	23.93	6549	24.39
150	9999	7570	24.29	7527	24.72
160	11430	8721	23.70	8675	24.10
170	12877	9881	23.27	9832	23.65
180	14533	11224	22.77	11166	23.17
190	16261	12462	23.36	12400	23.74
200	18163	13928	23.32	13860	23.69
220	21869	16855	22.93	16782	23.26
240	26109	20145	22.84	20051	23.20
260	30828	23855	22.62	23764	22.91
280	36235	28104	22.44	28009	22.70
300	41255	32078	22.24	31962	22.53
320	47153	36617	22.34	36484	22.63
340	53362	41378	22.46	41235	22.73
360	60281	46714	22.51	46562	22.76
380	67050	52369	21.90	52205	22.14
400	74467	58258	21.77	58073	22.02
420	82586	64563	21.82	64362	22.07
440	90378	70731	21.74	70520	21.97
460	98804	77236	21.83	77001	22.07
480	108033	84561	21.73	84317	21.95
500	117339	92054	21.55	91800	21.77
600	168910	132363	21.64	132024	21.84
700	231763	182698	21.17	182235	21.37
800	303586	238811	21.34	238273	21.51
900	383901	304147	20.77	303468	20.95
1000	474072	374947	20.91	374182	21.07

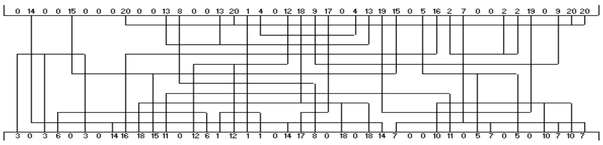
Table 1: Performance of our crosstalk reduction algorithms for two-layer *simplest* channel instances.

Number of Nets	Initial Crosstalk	After *Track Interchange*	Reduction after *Track Interchange* (%)	After *Net Change*	Reduction after *Net Change* (%)	After Greedy	Reduction after Greedy (%)	After *Net Change*	Reduction after *Net Change* (%)
20	135	125	7.41	118	12.59	127	5.93	120	11.11
40	564	530	6.03	494	12.41	540	4.26	499	11.52
60	1363	1291	5.28	1202	11.81	1312	3.74	1214	10.93
80	2475	2347	5.17	2183	11.80	2386	3.60	2198	11.19
100	3997	3808	4.73	3534	11.58	3875	3.05	3568	10.73
150	9042	8626	4.60	8028	11.21	8787	2.82	8100	10.42
200	16260	15498	4.69	14395	11.47	15851	2.52	14516	10.73
250	25293	24174	4.42	22492	11.07	24648	2.55	22658	10.42
300	36612	35004	4.39	32577	11.02	35760	2.33	32824	10.35
350	49708	47552	4.34	44228	11.02	48596	2.24	44546	10.38
400	65032	62164	4.41	57691	11.29	63666	2.10	58105	10.65
450	82857	79077	4.56	73323	11.51	81039	2.19	73881	10.83
500	102922	98324	4.47	91530	11.07	100788	2.07	92086	10.53
600	148035	141247	4.59	131355	11.27	144816	2.17	132193	10.70
700	201940	192513	4.67	179071	11.32	197677	2.11	180150	10.79
800	265574	252957	4.75	235523	11.32	259729	2.20	236978	10.77
900	336455	320481	4.75	298856	11.18	329553	2.05	300532	10.68
1000	412991	393305	4.77	366015	11.37	404210	2.13	368171	10.85
1500	938379	890801	5.07	831324	11.41	918524	2.12	836159	10.89
2000	1665471	1579930	5.14	1468851	11.81	1624401	2.47	1471997	11.62

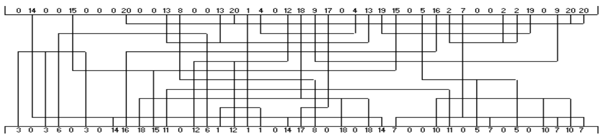
Table 2: Performance of our crosstalk reduction algorithms for two-layer *general* channel instances.



(a)

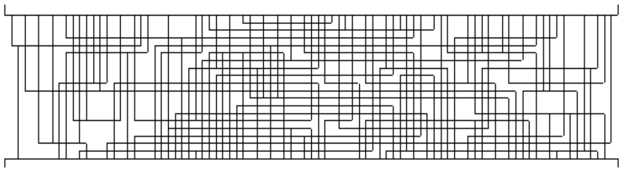


(b)

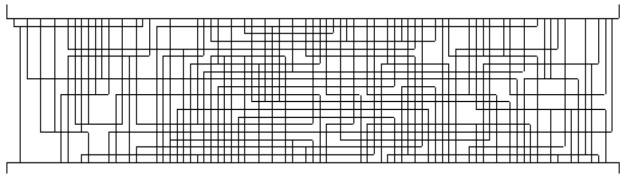


(c)

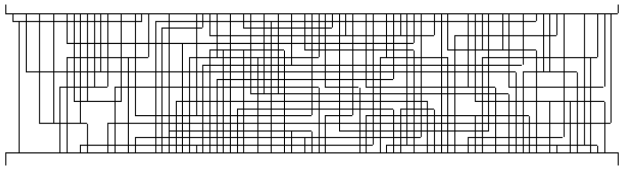
Figure 2: **(a)** The initial crosstalk is 176 units, **(b)** Crosstalk after algorithm *Track Interchange* is 149 units, and **(c)** after algorithm *Net Change* is 131 units only.



(a)



(b)



(c)

Figure 3: (a) The initial crosstalk is 680 units, **(b)** Crosstalk after algorithm *Track Interchange* is 578 units, and **(c)** after algorithm *Net Change* is 547 units only.

5. Conclusion

In this paper we have developed several heuristic algorithms for computing reduced crosstalk routing solutions for two-layer channel routing, and executed each of them on hundreds of randomly generated *simplest* as well as *general* channel instances. Results show that percentage reduction in crosstalk for smaller to larger *simplest* channel instances vary from approximately 35% to 21% and that of almost all *general* channel instances is roughly constant at 11-12% on the average. Algorithms for reducing crosstalk for the three-layer HVH routing model could be devised as an extension of the work presented here.

REFERENCES

1. Deutsch D. N., A Dogleg Channel Router, Proc. of 13th ACM/IEEE Design Automation Conf., pp.425-433, 1976.
2. Golumbic M. C., Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.

3. Hashimoto A. and J. Stevens, Wire Routing by Optimization Channel Assignment within Large Apertures, Proc. of 8th ACM Design Automation Workshop, pp. 155-169, 1971.
4. Ho T.-T., S. S. Iyengar and S.-Q. Zheng, A General Greedy Channel Routing Algorithm, IEEE Trans. on CAD 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. Pal A., A. K. Datta and R. K. Pal, Crosstalk Minimization in Two-Layer Channel Routing, manuscript, 2006.
7. 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.
8. 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 8th Annual Int. Conf. on Computer, Communication, Control and Engineering, vol. 1, pp. 569-573, 1993.
9. Rivest R. L. and C. M. Fiduccia, A 'Greedy' Channel Router, Proc. of 19th ACM/IEEE Design Automation Conf., pp. 418-424, 1982.
10. Szymanski T. G., Dogleg Channel Routing is NP-Complete, IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 4, pp. 31-41, 1985.
11. Yoshimura T. and E. S. Kuh, Efficient Algorithms for Channel Routing, IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 1, pp. 25-35, 1982.