

AN EFFICIENT APPROACH FOR FOUR-LAYER CHANNEL ROUTING IN VLSI DESIGN

Ajoy Kumar Khan¹, Bhaskar Das² and Tapas Kumar Bayen³

¹Department of Information Technology, Assam University, Silchar, India
ajoyiitg@gmail.com

²Department of Information Technology, Assam University, Silchar, India
bhaskardas2006@gmail.com

³Department of Computer Science and Engineering, N. I. S. T, Berhampur, India
tapas.bayen@gmail.com

ABSTRACT

Channel routing is a key problem in VLSI physical design. The main goal of the channel routing problem is to reduce the area of an IC chip. If we concentrate on reducing track number in channel routing problem then automatically the area of an IC chip will be reduced. Here, we propose a new algorithm to reduce the number of tracks using four layers (two horizontal layers and two vertical layers). To be more specific, through this algorithm we convert a two-layer channel routing problem into a four-layer channel routing problem using VCG of the channel. Next, we show the experimental results and graphical structure of that solution.

KEYWORDS

Track, Channel routing, Manhattan routing model, VCG & Merging.

1. INTRODUCTION

We all know that routing is very important problem in VLSI physical design. The main goal of routing is the reduction of area of an IC chip. The process of finding the geometric layout of all nets is called routing. Nets must be routed within the routing regions. In addition, nets must not short-circuit, i.e. nets must not intersect each other.

The input of the general routing problem is:

1. Netlist.
2. Timing budget of nets, typically for critical nets only.
3. Placement information including location of blocks, location of pins on the block boundary as well as on top due to ATM model (sea-of-pins model), and location of I/O pins on the chip boundary as well as on top due to C4 solder bumps.
4. RC delay per unit length on each metal layer, as well as RC delay for each type of via.

The traditional approach to routing, however, divides the routing into two phases. The first phase is called Global Routing and second phase is called Detailed Routing. In Global Routing, we generate a loose route for each net. In fact it assigns a list of routing regions to each net without specifying the actual geometrical layout of wires. In Detailed Routing we find actual geometric layout of each net within the assigned routing regions. We are in global routing phase.

To make easier to the routing problem the routing regions are divided into rectangular blocks. The blocks contain pins which need to be connected in their perimeter. There are two types of routing regions: Channel and switchbox. We are concentrating on channel routing problem. A channel is rectangular area which has two open ends and other two sides are bounded by two opposite sides of the blocks.

Channel routing has been used to implement the layout of Integrated Circuits (ICs) in a variety of design styles such as gatearrays, standard cells, and macro-cells (building blocks). The traditional channel routing problem assumes that two layers of material are available for routing. Advances in manufacturing technology, however, have made it practical to use three or even four layers for interconnections, and it is possible that in the near future more interconnect layers will be available.

We know that in Manhattan routing model we can only use horizontal wire segments and vertical wire segments to interconnect the nets. The layer with only horizontal wire segments is called horizontal layer (H) and the layer with only vertical wire segments is called vertical layer (V). In the case of multi-layer we place the horizontal and vertical layers alternatively. The connection between the layers is done through via hole.

Now we are going to discuss some important concepts which are needed in our proposed algorithm. These are HCG, VCG, HNCG and transitive HNCG. HCG is an undirected graph in which each node represents the net. In HCG any vertex adjacent to some other vertex means that these two nets can't be given to the same track. VCG is directed graph in which the directed edge represents that the track occupied by the source vertex or net must lie above the track occupied by the destination vertex or net. HNCG is the complement of HCG. The figure 1 shows an example of a channel and its HCG, VCG and HNCG of that particular channel.

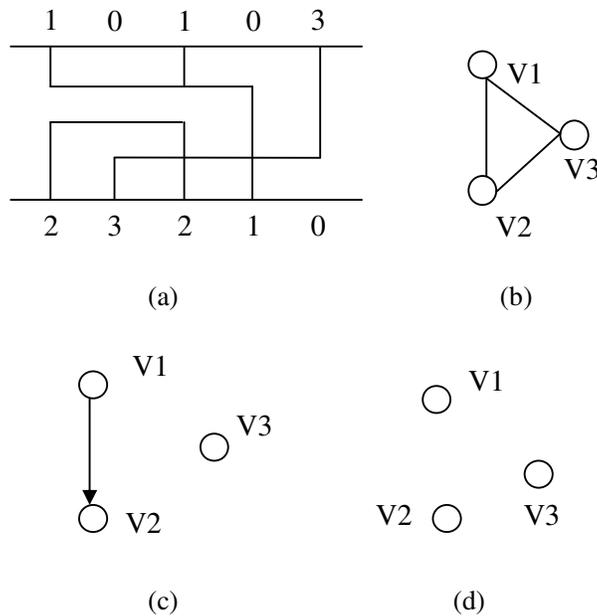


Figure 1. (a) An example channel (b) HCG (c) VCG (d) HNCG of the channel.

In this paper, we are transforming the two layer channel routing problem into a four layer channel routing problem using a new approach. Next part we discuss about problem formulation and next

we discuss the algorithm using an example. After that we show the result and future scope of our newly developed algorithm.

2. RELATED WORKS

In 1971, Hashimoto and Stevens [12] proposed an algorithm, for routing channels without vertical constraints. Their algorithm, commonly known as the left edge algorithm (LEA), can route a channel using no-dogleg routing within density number of tracks. The LEA is an iterative algorithm. In each step it computes a set of pair wise non-overlapping intervals for assignment to a track. Such a set is computed in each step starting from the leftmost net in the remaining channel. Consider the first iteration. Let I_1 be the interval with a terminal at the leftmost column position. Let I_2 be the interval with leftmost starting column position after the right end of I_1 . Non-overlapping intervals I_1 and I_2 are selected by the LEA for assignment to the first track.

Yoshimura and Kuh [7] proposed a different but very efficient algorithm for two layer channel routing. They divide the channel area into many zones first. This Algorithm doesn't allow the dogleg routing. This algorithm is based on the merging technique of two nets. They considered that two nets can be merged if there is no horizontal overlapping and there is no direct path in the vertical constraint graph between these two nets. We scan from left zone to right zone. We check that in one zone one net ends and one net begins from the next zone. If these two nets have one directed path in vertical constraint graph then we choose the net from previous zone to merge with it. Otherwise we merge these two nets. After merging we also update the VCG accordingly. This procedure is continuing to the last zone.

Yoshimura and Kuh [7] again proposed another algorithm which is the variant of the above algorithm. In that algorithm, the pair of nets to be merged is selected using bipartite graph matching technique.

Here we analyse two algorithms: MCC1 [6] and MCC2 [6]. Here we consider that there is no vertical constraint. So we resolve the horizontal constraint to get the minimum track needed for routing. The first algorithm MINIMUM_CLICK_COVER_1 (MCC1) is based on graph theoretic approach. The second algorithm MINIMUM_CLICK_COVER_2 (MCC2) is based on balanced binary search tree data structure.

Then one new algorithm is proposed by Chen and Liu [9] and is applicable in three-layer channel routing. This algorithm is extension of net merging method. This router separately considers routing for the VHV and HVH routing models. In VHV case the Left Edge Algorithm (LEA) is extended. In this case, after assigning all the horizontal wire segments to the horizontal layer using LEA, all upward vertical wire segments are assigned to one vertical layer and all downward vertical layer segments are assigned to the another vertical layer. In the case of HVH, this algorithm takes the concept of merging techniques. Here, the router not only merges the nets between two different zones (as in two-layer routing) but also merges the nets in the same zone. The merging of nets between two zones is called serial merging and the merging of nets in the same layer is called parallel merging.

New techniques for routing general multi-layer channels are introduced by D. Braunt, J. Burns, S. Devadas, H. K. Ma, K. Mayaram, F. Romeo, and A. Sangiovanni-Vincentelli [13]. These techniques can handle a variety of technology constraints. For example, linewidth and line-to-line spacing can be specified independently for each layer, and contact stacking can be allowed or forbidden. These techniques have been implemented in a new multi-layer channel router called Chameleon.

Greenberg, Ishii and Sangiovanni-Vincentelli [3] proposed an algorithm MulCh for routing a channel which has more than four layers. MulCh can use unrestricted doglegging in the reserved

layer Manhattan routing model and always route channels with cyclic vertical constraints. Here, the problem is decomposed by assigning the nets into groups. One is VH group, another is HVH group and another new group called B group. MulCh allows the layer called B layer on which wire segments are allowed to run both horizontal and vertical directions.

3. PROBLEM STATEMENT

The primary objective of routing is to minimize the channel area. The secondary objectives are to minimize the number of vias and to minimize the length of each net. The objectives may be in opposition: for example, doglegging can reduce area but it can also increase net length and via count.

If we concentrate on reducing the number of tracks needed for connecting the nets then automatically the area is also reduced. Here we use two horizontal layers and two vertical layers and trying to find out minimum number of tracks for connecting the nets.

4. PROPOSED ALGORITHM

In this section, we propose a new algorithm for the four layer channel routing problem where there is no cycle in VCG of that channel. To be more specific, through this algorithm we transform the two layer routing problem into four layer routing problem using VCG of that channel. Here we use two horizontal layers and two vertical layers. In our proposed algorithm, the horizontal segments of which nets are placed in first horizontal layer, the vertical segments of those nets are paced in first vertical layer and the same is happened in second horizontal layer and second vertical layer also. First, we draw the vertical constraint graph (VCG) of a given channel. Then, all the nets in the channel are sorted according to their starting column position in ascending order.

Let us take an example to illustrate our algorithm. Figure 2 shows an example channel. Figure 3 shows the VCG of that channel. In this example channel we can see that the net 7 starts from first column and next net 2. So, after sorting all the nets, the sorting order of the nets becomes (7, 2, 1, 5, 4, 3, 8, 6).

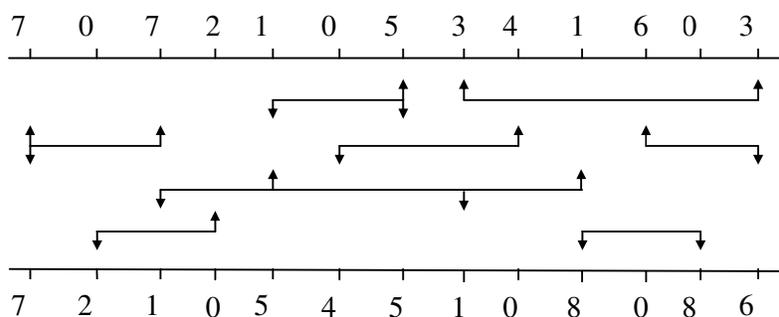


Figure 2. An Example Channel for Routing

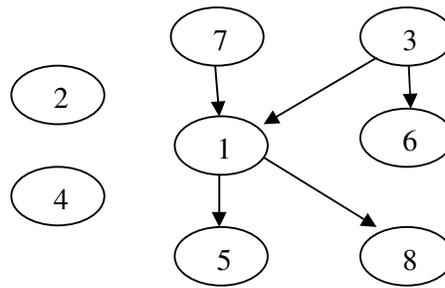


Figure 3. VCG of figure 2.

First we select that net N which is first in the sorted list and that is stored into another list L . Then we choose the net N_s which start first after the net N ends. Then we check if there is a path or not between the any nets in the list L and N_s . If there is a path, then left N_s and choose the net which starts next and that is the new N_s . Next N_s becomes N and this is also inserted into L . This process is continued till the columns of channel ends. These nets which are in list L that is placed into one track. Then, the nets which are in list L those nets are deleted from the sorted list and merge these vertices in VCG and that becomes the new VCG. Then again select the first net in sorted list. This whole process repeats till the sorted list is empty.

In the figure 2 we can see that 7 is the first net in the sorted list and so we select 7. After 7 ends, 5 starts first but there is a path between 7 and 5 in VCG. So, we choose the net which starts next and that is 4. We can see that there is no path between 7 and 4 and so we select 4. Then, again after 4 ends, 8 starts first but there is a path between 7 and 8. So, we choose next and that is 6. We can see that there is no path between 7 and 6 or 4 and 6. So we select 6. So, one L becomes (7, 4, 6) and these nets can be placed into one track. Then Merge these vertices in VCG and that become the new VCG shown in figure 4. Next delete these nets from sorted list. Now again select the first net in sorted list and repeat these all steps until the sorted list become empty. According to the algorithm the next L becomes (2, 5, 8) and merge this vertices in new VCG shown in figure 5. Two other lists are (1) and (3).

So we need four tracks in two layer channel routing problem. Here we use two horizontal layers. So we need minimum two tracks for routing. The first two cliques are placed into first horizontal layer and next two cliques are placed into second horizontal layer. And vertical wire segments for first two cliques are placed into first vertical layers and vertical wire segments of next two cliques are placed into second vertical layer. So we are using four layers (HVHV) for routing.

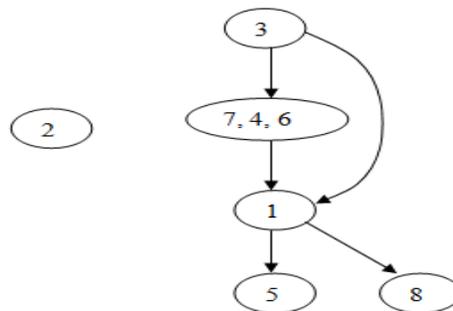


Figure 4. Merged VCG after First List

Now, the lists (7, 4, 6) and (2, 5, 8) are placed in the first horizontal layer and the lists (1) and (3) are placed in the second horizontal layer. We can see that (7, 4, 6) are on the upper level than (2, 5, 8). So, (7, 4, 6) are placed on first track and the (2, 5, 8) is placed on the second track. Similarly for the second horizontal layer (3) is placed on first track and (1) is placed on second track.

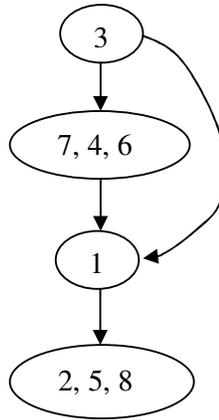


Figure 5. Merged VCG after Second, Third and Last List

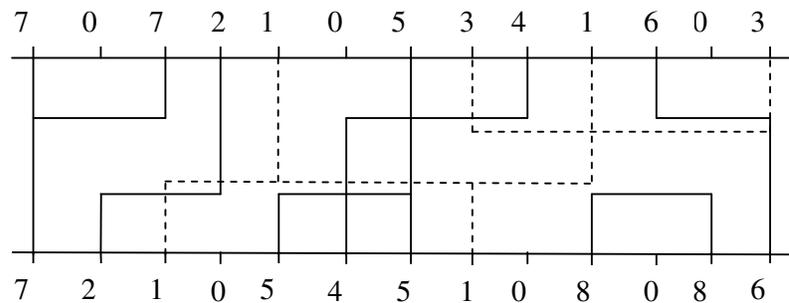


Figure 6. Solution of figure 2.

In our example, the number of lists is 4 and that is even. But if the number of lists is odd then first $n/2$ lists are placed in first horizontal layer and the rest are placed in the second horizontal layer where n is the number of lists. So, now the figure 6 shows the solution of the example channel shown in figure 2. The nets which are drawn by plain lines are placed in first horizontal and vertical layers, and the nets which are drawn by dotted lines are placed in second horizontal and vertical layers.

Now we are in the algorithm part.

Algorithm: Minimum_Track_1

Input: Channel Specification.

Output: Minimum number of tracks for four layers.

Begin

Step 1: Sort the nets according to their starting column positions in ascending order.

Step 2: Construct the VCG $G = (V, E)$. Take a variable count = 0.

Step 3: while (the sorted list is non-empty) do

begin

Step 3.1: Assign a list $L \leftarrow \Phi$.

Step 3.2: Select the first net N from the sorted list. Assign $L \leftarrow L \cup N$.

Step 3.3: Repeat.

Step 3.3.1: Select the net N_s which starts first after N ends

Step 3.3.2: Repeat.

Step 3.3.2.1: Check in that there is one path or not between N_s and any vertex in the list L in VCG.

Step 3.3.2.2: If path exists then left N_s and choose which starts next N_s and that is new N_s .

Step 3.3.3: until one net is selected or Scan all the nets.

Step 3.3.5: Now assign $N \leftarrow N_s$ and $L \leftarrow L \cup N$.

Step 3.4: Until the vertex N is a last net.

Step 4.5: L is one list and count++.

Step 4.6: Merge L vertices in VCG and it is new VCG.

End

Step 4: Minimum number of tracks $T_s = \lceil \text{count}/2 \rceil$

End

5. EXPERIMENTAL RESULTS

The program is written in C programming language in Windows XP platform. The result is very good. We show the results in table 1. These examples are taken from [9].

The figure 7 shows the graphical representation of Ex.6. We can see that in two-layer problem it need 8 tracks for routing but in our four-layer channel routing algorithm we need only 4 tracks. In the same way figure 8 and figure 9 also shows the solution of Ex. 7 and Ex. P&G. So, we can come to know that the new algorithm gives the better result comparison to any two layer channel routing algorithms.

Table 1. Experimental Results.

Examples	No. of nets	No. of tracks (Two-Layer)	No. of tracks (Four-Layer)	% of reduction
Ex. 1	5	4	2	50%
Ex. 2	6	6	3	50%
Ex. 3	7	5	3	40%
Ex. 4	8	5	3	40%
Ex. 5	10	5	3	40%
Ex. 6	12	8	4	50%
Ex. 7	19	8	4	50%

RKPC3	6	4	2	50%
RKPC7	8	7	4	42.86%
RKPC8	10	7	4	42.86%
Ex. P&G	21	12	6	50%
Example2	30	16	8	50%

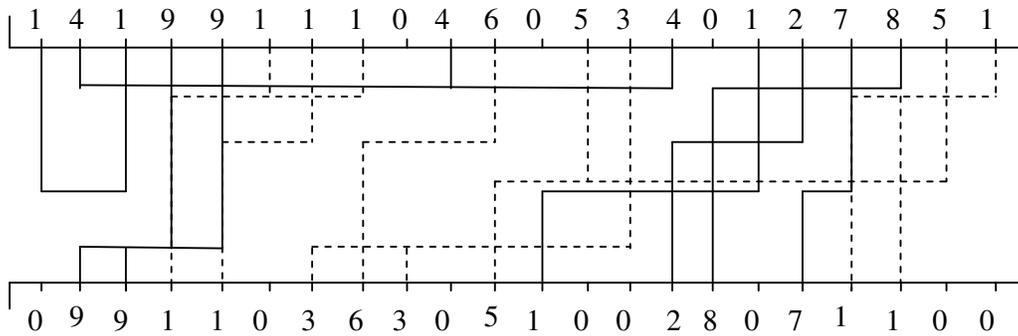


Figure 7. Solution of Ex. 6.

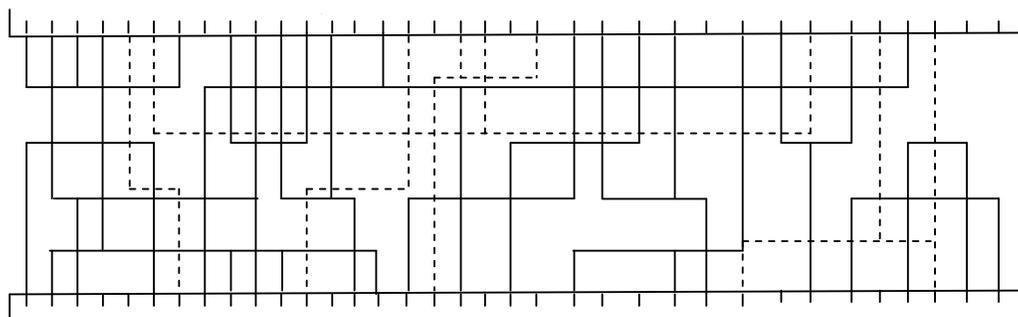


Figure 8. Solution of Ex. 7

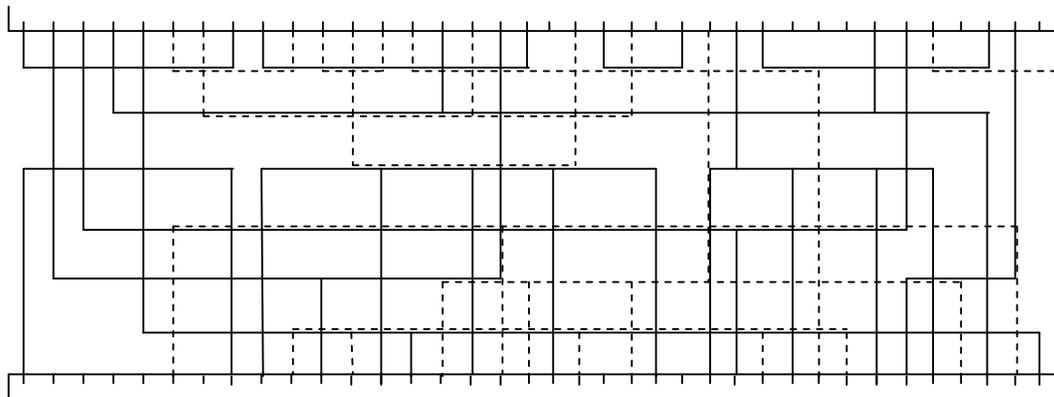


Figure 9. Solution of Ex. P&G

6. CONCLUSION AND FUTURE SCOPE

So, we are in the end of this paper. Here we reduced the number of tracks needed for channel routing problem. We know that if we reduce the number of tracks then automatically the area of an IC chip is also reduced. So, we can conclude that our algorithm gives very good result comparison to any two-layer channel routing problem but our algorithm will run only when there is no close loop in the initial VCG of that channel that means the VCG must be a tree.

We know that if there is a cycle in VCG then we must need one more vertical layer. So, we can add one more vertical layer before horizontal layer. Then it becomes a five-layer channel problem. Then again another problem arises that if there are more than two cycles in VCG then what to do? So, we have to draw some algorithm for this purpose also. That is our future scope.

REFERENCES

- [1] A. Pal, A. K. Khan, S. Saha Sau, A. K. Dutta, R. K. Pal and A. Chaudhury, (2010) "Application of Graph in Computing Reduced Area VLSI Channel Routing Solutions," *Proc. Of International Conference on Computing and Systems (ICCS- 2010)*, pp. 249-256.
- [2] A. Pal, T. N. Maldal, S. Saha Sau, A. K. Dutta, R. K. Pal and A. Chaudhuri, (2011) "Graph – The Tool to Visualize the Problems in VLSI Channel Routing." *Assam University Journal of Science & technology*, Vol. 7, Number II, pp 73-83.
- [3] R. I. Greengerg, A. T. Ishii and A. L. Sangiovanni –Vincentelli, (1988) "Mulch: A Multi-Layer Channel Router Using One, Two, and Three Layer Partitions," *IEEE International Conference on Digital Object Identifier*, pp. 88-91.
- [4] R. K. Pal, A. K. Datta, S. P. Pal, M. M. Das and A. Pal, (1995) "A General Graph Theoretic Framework for Multi-Layer Channel Routing," *Proc. Of Eighth VSI/IEEE International Conference on VLSI Design*, New Delhi, India, pp. 202-207.
- [5] A. Pal, T. N. Maldal, S. Saha Sau, A. K. Dutta, R. K. Pal and A. Chaudhuri, (2011) "An Efficient Heuristic to Find Reduced Area VLSI Channel Routing Solutions with Floating Terminals" *Assam University Journal of Science & technology*, Vol. 9, Number II, pp 55-64.
- [6] R. K. Pal, A. K. Datta, S. P. Pal and A. Pal, (1993) "Resolving Horizontal Constraints and Minimizing Net Wire Length for Multi-Layer Channel Routing," *Proc. Of IEEE Region 10's 8th Annual Int. Conj. On Computer, Communication, Control and Engineering (TENCON 1993)*, vol. 1, pp. 569-573.
- [7] T. Yoshimura and E. S. Kuh, (1982) "Efficient Algorithms for Channel routing," *IEEE Trans. On CAD of Integrated Circuits and Systems*, vol. 1, pp. 25-35.
- [8] R. K. Pal, (2000) "Multi-Layer Channel Routing: Complexity and Algorithms" *Narosa Publishing House*, New Delhi.
- [9] Y. K. Chen and M. L. Liu, (1984) "Three-Layer Channel Routing," *IEEE Trans. On CAD*, Vol. Cad-3, No.2, pp. 156-163.
- [10] R. K. Pal, S. P. Pal and A. Pal, (1998) "An Algorithm for Finding a Non-Trivial Lower Bound for Channel Routing," *INTEGRATION: the VLSI Journal*, Vol. 25, No. 1, pp. 71-84.
- [11] J.Reed, A.Sangiovanni–Vincentelli and M. Santomauro, (1985) "A New Symbolic Channel Router: YACR2," *IEEE Trans. On CAD of Integrated Circuits and Systems*, vol. 4, pp. 208-219.
- [12] Hashimoto A. and I. Stevens, (1971) "Win routing by optimiaing channel assignment within large apertures", *Proc. of 8th Design Aufomation Wombhop*, pp. 155-169.
- [13] D. Braunt, J. Burns, S. Devadas, H. K.Ma, K. Mayaram, F. Romeo, and A. Sangiovanni-Vincentelli, (1986) "Chameleon: A New Multi-Layer Channel Router," *IEEE 23rd Design Automation Conference*, pp. 495-502.
- [14] P. Bruell and P. Sun, (1984) "A "Greedy" Three Layer Channel Router". *Pmt. ZCCAD-85*. pp. 298-300.
- [15] D. Deutsch, (1976) "A 'dogleg' channel router". *A-cc. 23th Design Automation Conf.*, pp. 425-433.
- [16] R. Rivest, (1982) "Benchmark channel-routing problems", *private communication*.

Authors

Ajoy Kumar Khan was born on 17th September, 1979 in Kolkata, India. He completed his M.Tech degree from Calcutta University, Kolkata, in 2007, in Computer Science and Engineering. Now, he is pursuing Ph.D in the area of VLSI Design from Assam University, Silchar, India. Presently, he is serving as Assistant Professor in the Department of Information Technology, Assam University. He has more than 5 years teaching experience. His main research interests are in the area of VLSI Design and Network Security.



Bhaskar Das was born in Kolkata, India on 12th June, 1987. He graduated from West Bengal University of Technology, Kolkata, in Computer Science and Engineering, in 2009. Then he was a Lecturer in the department of Information Technology in Uttar Pradesh Technical University from 2009 to 2011. Now, he is pursuing M.Tech in Information Technology from Assam University, Silchar, India. His main research interests are in the area of VLSI Design and Design of Algorithm.



Tapas Kumar Bayen was born in Kolkata, India on 10th April, 1980. He completed his M.Tech degree from Calcutta University, Kolkata, in 2008, in Computer Science and Engineering. Presently, he is serving as Assistant Professor in the Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, Orissa, India. He has more than 5 years teaching experience. His main research interests are in the area of VLSI Design and Algorithm.

