A Review on Channel Routing On VLSI Physical Design

Ajoy Kumar Khan¹, Bhaskar Das², Tapas Kumar Bayen³

^{1, 2}Information Technology, Assam University, Silchar, India

³Computer Science and Engineering, National Institute of Science and Technology, Orissa, India

Abstract: We know that channel routing is very important problem in VLSI physical design. The main objective of a channel routing algorithm is the reduction of the area of a IC chip. In this paper, we just do a survey on some impotent multi-layer routing algorithms. Here we analyze The Efficient Routing algorithm and resolve horizontal constraints and minimize the net wirelength in a particular model of channel routing using MCC1 and MCC2 algorithms. Next, we analyze an algorithm for Multi channel Routing MulCh and its differences from Chameleon which s another multi channel routing in the two-layer VH and three-layer HVH routing models. **Keywords:** Multi-layer channel routing Manhattan routing model, Channel density, Minimum clique, Track.

I. INTRODUCTION

Before going to the main topic we have to know about routing. We can view the spaces which are not occupied by the blocks as a collection of regions. These regions are used for routing. These regions are called routing regions. The process of finding the geometric layouts of all the nets is called routing. It has two phases. First phase is global routing. In this phase we generate a 'loose' route for each net. The second phase is called detailed routing. In this phase we find the actual geometric layout of each net within the assigned routing regions.

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. A channel is rectangular area which has two open ends and other two sides are bounded by two opposite sides of the blocks. Switchbox can be two types: 2D-switchbox and 3D-switchbox. A 2D-switchbox is rectangular area bounded by three or four sides of the blocks. A 3D-switchbox is the rectangular area bounded by the blocks. Here we focus only the channel routing.

In 3D, a channel is a rectilinear grid with t+2 rows, m+2 columns and z layers [3]. A 3D view of a channel is shown in the figure 1.1. This channel has 6=t+2 rows, 10=m+2 columns and 4 layers. Rows are numbered as 0 to t+1 and columns are numbered as 0 to m+1. The 0th and $(m+1)^{th}$ columns and the 0th and $(t+1)^{th}$ rows are not used to make connections inside the channel but used to make connections to circuits outside the channel. The intersection between layer, row and column is called grid point GP[h, j, k] where h=row no., j=column no. and k=layer no.



Figure 1.1: 3D view of a channel

We know that in Manhattan routing model we can only use horizontal wire segment and vertical wire segments to interconnect the nets. In grid-based Manhattan routing, horizontal wire segments are assigned to the different rows and vertical wire segments are assigned to the different columns. 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. In figure 1.2 shows 3D view of multi-layer.



If the number of layer is even then there must be same number of vertical and horizontal layers. This model is called V_iH_i model. If the number of layers is odd then there are two possible models. First model is V_iH_{i+1} model. In this model, first layer is horizontal layer and next vertical and next again horizontal layer and so on. Second model is $V_{i+1}H_i$ model. In this model first layer is vertical layer and next horizontal and next again vertical layer and so on.

Now we are going to discuss some important concepts. These are horizontal constraint graph (HCG) [1], vertical constraint graph (VCG) [1] and horizontal non-constraint graph (HNCG) [1]. Suppose we have a routing solution of a channel is shown in the figure 1.3.



HCG is a perfect 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 each node represents the net. The directed edge of vertical constraint graph represents that the track occupied by the source vertex or net must lies above the track occupied by the destination vertex or net. HNCG is the complement of HCG. The figure 1.4 shows the HCG, VCG and HNCG of the solution shown in figure 1.3.



Figure 1.4: (a) Horizontal constraint graph (b) Vertical constraint graph (c) Horizontal non-constraint graph

II. TWO LAYER CHANNEL ROUTING ALGORITHMS

2.1. Efficient Routing Algorithm

Yoshimura and Kuh [8] proposed a different but very efficient algorithm for two layer channel routing. Before going to the main algorithm we have to know the concept of zone representation of a channel. In a channel there are many columns. The zone representation is the classification of columns into different zones. Now the main question is how the classification is done. We scan from left column to right column. Suppose ith column is in one zone say zone Z. We check, if in ith column at least one net ends and in $(i+1)^{th}$ column at least one new net begins, then we begin a new zone (Z+1) from (i+1) column. Otherwise $(i+1)^{th}$ column is also in zone Z.

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. Here the figure 2.1 shows an example of a channel and its HCG. The table 2.1 shows the column number and the set of nets which intersects that column and its zone divisions.



Figure 2.1: (a) A channel [8] (b) HCG of this channel [8]

Columns	Nets	Zone
1	2	1
2	1, 2, 3	
3	1, 2, 3, 4, 5	
4	1, 2, 3, 4, 5	
5	1, 2, 4, 5	
6	2, 4, 6	2
7	4, 6, 7	3
8	4, 7, 8	4
9	4, 7, 8, 9	
10	7, 8, 9	
11	7, 9, 10	5
12	9, 10	

Table 2.1: Division of zones [8]



8



Figure 2.2: (a) The Zone representation [8] (b) Initial VCG [8] (c)Merging of nets according to the algorithm [8]



Figure 2.3: Final solution of figure 2.1 (a) [8]

In the VCG, we consider the merged nets as a single net that means we can place these nets in one track. Figure 2.3 gives the solution of figure 2.1 (a). This solution is based on the final VCG of figure 2.2 (c).

Yoshimura and Kuh [8] 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. Both the algorithms give better result than the dogleg [12] channel routing.

2.2. Algorithms for Two-Layer Routing in the Absence of Vertical Constraints

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. It takes O (n+ e) time to run where n = number of nets and e = size of HNCG. The second algorithm MINIMUM_CLICK_COVER_2 (MCC2) is based on balanced binary search tree data structure. In takes O (n log n) time to run.

Before going to the main algorithm we have to know about transitively oriented graph. We know that an undirected graph can be converted into a directed graph by orienting the directed edge. If there exists an orientation of all edges such that for any three vertices v_i , v_j and v_k : $v_i \rightarrow v_j$ and $v_j \rightarrow v_k$ implies $v_i \rightarrow v_k$, then that undirected graph is called transitively orientable graph. So, we can see that a transitively oriented graph has three types of vertex: source, sink and intermediate vertex. Source vertex has no incoming edges and sink vertex has no outgoing edges and intermediate vertex has both incoming and outgoing edges.

In HNCG, two vertex has a common edge means these two nets do not overlap each other. So we can see that a clique in HNCG says that nets can be placed into one track. So, here we calculate the minimum click cover of HNCG of the channel. The figure 2.4 shows the interval and HNCG of a channel.



Figure 2.4: (a) Intervals of channel [6] (b) HNCG of the channel [6]

Now each vertex of the HNCG is given a particular number say key value. The main question is how we will give that key value. We scan the channel from the left to right. That interval which starts first, that net or that

vertex is assigned the number 1 next is assigned number 2 and next is 3 and so on. Then we convert this undirected graph to a directed graph i.e. transitively oriented graph by giving the direction towards the higher number. The figure 2.5 shows the transitively oriented graph of figure 2.4.



Figure 2.5: Transitive graph of figure 2.4. [6]

First we sort the vertices according to the key value to ascending order. We select the first vertex. Then we look the destination vertices of only outgoing edges of the selected vertex. Then we select that vertex having the smallest key value. This process repeats till one sink vertex is selected. That becomes one clique. Then we delete these vertices from the transitively oriented graph. Then again sorting of vertices is done in the new graph. This process repeats till all the vertices are selected and the graph become empty. In figure 2.5, first we select v_5 then v_2 and then v_7 . As v_7 is a sink vertex, (v_5, v_2, v_7) become a clique and this means that the net 5, 2 and 7 can be placed into one track. In the same way, (v_3, v_1, v_4) and (v_6) is became another two cliques of the figure 2.5. We need minimum three tracks for routing of the channel given in figure 2.4.

Here we analyze another algorithm MINIMUM_CLICK_COVER_2 (MCC2) which is identical to MCC1 except the selection method. Here, first we select first interval from left in the channel. Next we select the interval which starts first after the ending of the selected intervals. These intervals are placed into one track. This computation can be done by maintaining a balanced binary search tree of intervals sorted according to their starting column number.

III. THREE-LAYER CHANNEL ROUTING ALGORITHM

This algorithm is proposed by Chen and Liu [10] 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. The serial merging of two nets is possible if there is neither horizontally nor vertically constrained and parallel merging of two overlapping nets can be possible if there is no directed path in the VCG. Figure 3.1 shows an example channel of HVH model.



Figure 3.1: An example channel [10]



Figure 3.2: Step by step merging of nets according to algorithm [10].

Figure 3.2 shows the step by step merging of nets. We know that here merging may be serial or parallel. The serial merged nets can be placed into one track but parallel merged nets can't. So we divide the horizontal wire segments into two groups. One group is placed into one layer and another group is into another layer. Here in figure 3.2 the nets shown by dotted line are into one group and others are into another group. So the final routing is shown n he figure 3.3.



IV. MULTI-LAYER CHANNEL ROUTING

4.1. MulCh: A Multi-Layer Channel Routing Algorithm

Greenberg, Ishii and Sangiovanni-Vincentelli [9] 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.

MulCh has two major components, the partitioner and the detailed router. The partitioner determines the group types of layers and partitions nets between them. The detailed router completes the interconnections by assigning the wire segments in the channel. MulCh follows the strategy of Chameleon[11] once the number of B layer is set. Chameleon is able to choose the group types independently. The strategy of chameleon is to use as many HVH group as possible and up to two VH groups because usually the best results are obtained with largest possible number of H layers.

Using more B layer reduces the standard lower bound on channel width but an excessive number of B layers may prevent us from coming as close to this lower bound because of the additional constraints on net assignment and detailed routing. For this, we start with no B layer and estimate the minimum area required for the routing. Then we select a minimum number of B layer and again estimate the area. If there is any improvement then we choose second one. For example, if there are 7 layers then first we choose HVH, VH and VH groups and estimate the area required for routing. Next we choose HVH, HVH and B groups then estimate the area again. Next again we try HVH, B, B, B, B and then try seven B layers. Which one is the best we choose that one.

Each net is initially added to all the groups and then the cost is estimated. Then the net is removed from all the groups except the best one. In the detailed routing phase, the channel density and the length of the longest path is used to estimate the channel width for VH and HVH group. The consideration is different for B layer. First and foremost we must be sure that a proposed assignment of a net to a B layer does not yield a non-planar collection of nets in that group. This checking is done from the partitioning process. So, MulCh is an enhanced version of Chameleon.

4.2. Multi-Layer V_{I+1}H_i Channel Routing

In this section, we show how algorithm MCC1 and MCC2 can be applied to compute feasible solution for $V_{i+1}H_i$, $i\geq 1$ channel routing model [6]. The algorithm MCC1 and MCC2 give the feasible solution only if there is no vertical constraints in two-layer routing channel. But in general CRP there is both horizontal and vertical constraints. In order to route channels with cycles in the VCG we must we two or more vertical layers. Consider the VHV routing model. Any vertical constraint (v_i, v_j) can be resolved in this model by routing the vertical wire segments of the nets n_i and n_j using the two vertical layers on either side of the horizontal layer. So it is immaterial whether n_i is assigned to a track above n_j or n_j is assigned to a track above n_i . So, here also we just consider only horizontal constraints.

Now, we consider two or more horizontal layers in order to reduce the routing area. If we use i number of horizontal layers then we need at least $[\overline{d}_{max}/i]$ Tracks and also we need i+1 vertical layers. We first compute the minimum clique cover of HNCG using MCC1 or MCC2 algorithms. Then, the d_{max} cliques of the minimum clique cover are assigned arbitrarily to d_{max} tracks spread over i horizontal layers. After assigning the tracks we route the vertical layers as follows. Since there is a vertical layer on each side of a horizontal layer, we can route each of the two vertical wire segments using these two vertical layers.

V. CONCLUSION

We have discussed about some multi-layer channel routing algorithms and minimization of net wirelength in this paper. The merging algorithm used in two-layer routing can be extended to three layers. MCC1 and MCC2 are used only if there is no vertical constraint.

We have seen that the MulCh give the better result than the original chameleon. In many cases, Mulch succeeded in simultaneously reducing the number of rows and vias, as well as total net length. The computational complexity question remains open for minimizing the total routing wire length in the absence of any such restriction.

REFERENCES

- [1]. A. Pal, T. N. Maldal, S. Saha Sau, A. K. Dutta, R. K. Pal and A. Chaudhuri, "Graph The Tool to Visualize the Problems in VLSI Channel Routing." *Assam University Journal of Science & technology*, Vol. 7, Number II, pp 73-83, 2011.
- [2]. A. Pal, A. K. Khan, S. Saha Sau, A. K. Dutta, R. K. Pal and A. Chaodhury, "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.
- [3]. R. K. Pal, "Multi-Layer Channel Routing: Complexity and Algorithms," Narosa Publishing House, New Delhi, 2000.

- [4]. A. Pal, T. N. Maldal, S. Saha Sau, A. K. Dutta, R. K. Pal and A. Chaudhuri, "An Efficient Heuristic to Find Ruduced Area VLSI Channel Routing Solutions with Floating Terminals" Assam University Journal of Science & technology, Vol. 9, Number II, pp 55-64, 2011.
- [5]. R. K. Pal, A. K. Datta, S. P. Pal, M. M. Das and A. Pal, "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, 1995.
- [6]. R. K. Pal, 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. Conj. on Computer, Communication, Control and Engineering (TENCON 1993), vol. 1, pp. 569-573, 1993.
- [7]. R. K. Pal, S. P. Pal and A. Pal, "An Algorithm for Finding a Non-Trivial Lower Bound for Channel Routing," *INTEGRATION: the VLSI Journal*, Vol. 25, No. 1, pp. 71-84, 1998.
- [8]. T. Yoshimura and E. S. Kuh, "Efficient Algorithms for Channel routing," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 1, pp. 25-35, 1982.
- [9]. R. I. Greengerg, A. T. Ishii and A. L. Sangiovanni Vincentelli, "Mulch: A Multi-Layer Channel Router Using One, Two, and Three Layer Partitions," *IEEE International Conference on Digital Object Identifier*, pp. 88-91, 1988.
- [10]. Y. K. Chen and M. L. Liu, "Three-Layer Channel Routing," *IEEE Trans. on CAD*, Vol. Cad-3, No.2, pp. 156-163, 1984.
- [11]. D. Braunt, J. Burns, S. Devadas, H. K.Ma, K. Mayaram, F. Romeo, and A. Sangiovanni-Vincentelli, "Chameleon: A New Multi-Layer Channel Router," *IEEE 23rd Design Automation Conference*, pp. 495-502, 1986.
- [12]. T. G. Szymanski, "Dogleg Channel Routing is NP-Complete," IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 4, pp. 31-41, 1985.
- [13]. J.Reed, A.Sangiovanni–Vincentelli and M. Santomauro, "A New Symbolic Channel Router: YACR2," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 4, pp. 208-219, 1985.