

Design And Optimization Of Tracks For Channel Routing In Vlsi Physical Design

Vinuta H.¹, Prof. Pavan Kumar E.²

¹(Dept. of E&C, S.V.I.T., Bengaluru, India)

²(Dept. of E&C, S.V.I.T., Bengaluru, India)

Abstract: The VLSI design is divided into different stages to reduce complexity. The physical design is one among the stages of VLSI design. It is also further classified into different phase to ease the physical design. Channel routing is one of the key areas in VLSI physical stage. The routing of channel problem is considered in the further discussion. The channel routing has many objectives such as reducing vias, minimizing length of wire and reduction of area utilized for routing. The work here is related to minimization of area used for channel routing while considering the constraints of routing. The constraint graphs for placing horizontal and vertical segments are constructed. The cliques are obtained from these Vertical Constraint Graph (VCG) and Horizontal Non Constraint Graph (HNCG). The algorithm proposed and implemented uses four layers for routing the channel where two layers are used for placing horizontal segments and two layers are used for placing the vertical segments. The results obtained are compared with algorithm using two layers and efficiency of the four layer routing solution is proved.

Keywords: Channel Routing, Clique, HCG, HNCG, VCG, Track

I. Introduction

The demand for miniaturization of integrated circuits has increased the complexity of IC layout. The different techniques and algorithms are designed for various phases of VLSI design in order to reduce complexity and ease of development and maintenance. The system under design is represented logically which forms input for physical design. The optimized layout is got from physical stage which realizes logical representation of the system. The VLSI design is divided into different stages such as system specification, architectural design, functional design, logic design, circuit design, physical design, fabrication, packaging and testing for the ease of development.

The physical design phase of VLSI design [1] is further classified to ease of development. The partitioning stage forms sub circuits from the circuits which speeds the design and also helps to improve efficiency. The partitioned sub circuits lie in the given range and complexity of connections is reduced between these sub circuits. The output of partitioned stage is blocks whose size and specific position is planned in floorplanning and placement stage. The interconnection path between these blocks according to the netlist obtained from placement phase is done in routing stage. The total area is minimized and design rule violations are eliminated in last stage compaction.

The routing is done in two phases. The first phase global routing finds routing regions and wires are loosely routed without considering actual geometry. The detailed routing phase routes nets where wire length and width are considered. Channel routing and switchbox routing are two types of routing.

Channel routing is one among the key problem of physical design stage in VLSI. Its objective is attrition of channel area which in turn reduces IC area. The area of channel is reduced by minimizing the use of tracks for placing horizontal segments. Channel is formed by opposite sides of two blocks and other two sides are open. The top and bottom of channel have terminals which need to be connected. These terminals are the input for routing problem. The routing is done by using horizontal layer for placing horizontal segment and vertical layer for placing vertical segment.

II. Problem Definition

The physical design phase is one among the VLSI design and it is classified into different phases for ease of design. Routing is one of the stages in physical design and routing problem is solved. There are many objectives of channel routing but at once all objectives cannot be satisfied. There are different algorithms which aim at achieving one objective. The attrition of channel area by utilizing fewer tracks for routing channel is considered here. The algorithm discussed here finds tracks required for routing using four layers.

III. Proposed Design

The fewer tracks are obtained for channel routing using the algorithm proposed. The placement stage gives netlist which is the input of routing algorithm. The constraint graphs [2] are obtained from these netlist and cliques are formed. These cliques give the count of tracks necessary for routing.

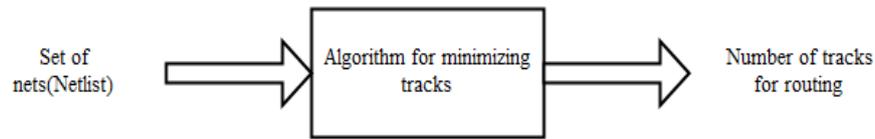


Fig.1. Block Diagram

The algorithm to connect the terminals by using fewer tracks is mentioned below.

- Step 1.** The set of terminals are sorted and keys are assigned to them based on their position on column.
- Step 2.** The constraint graphs such as HCG, HNCG and VCG are constructed. The orientation for HNCG is done which is transitive in nature. The variable track count = 0.
- Step 3.** Repeat below steps till transitive oriented graph is not empty.
 - Step 3.1.** ArrayList clique = null is created.
 - Step 3.2.** The first vertex from sorted nets is taken.
 - Step 3.3.** If selected vertex is not present in clique cover then adds it to arraylist clique.
 - Step 3.4.** Repeat steps below till empty sorted list is obtained.
 - Step 3.4.1.** The vertices obtained from transitive oriented graph corresponding to vertex selected are sorted in ascending order according to their column position.
 - Step 3.4.2.** Repeat below steps until list of sorted vertices does not become empty.
 - Step 3.4.2.1.** Select starting vertex from sorted vertices.
 - Step 3.4.2.2.** Verify if there is any path between vertices in clique and vertex selected in VCG.
 - Step 3.4.2.3.** Next vertex from sorted vertices is considered if there is any path.
 - Step 3.4.2.4.** Clique is updated with vertex selected if no direct path.
 - Step 3.5.** Clique cover is updated with the cliques obtained.
 - Step 3.6.** Increment track count.
 - Step 3.7.** VCG is updated by merging vertices involved in clique formation.
 - Step 3.8.** The transitive oriented graph is updated by deleting vertices and their respective edges involved in clique formation.
- Step 4.** Track count/2 gives the tracks for routing.

IV. Implementation

The algorithm is implemented using Java 1.7 and the tool used is Eclipse 4.5. The below class diagram shows the interfaces, classes and methods involved in the implementation. It also helps to understand the relation between the interfaces and classes used.

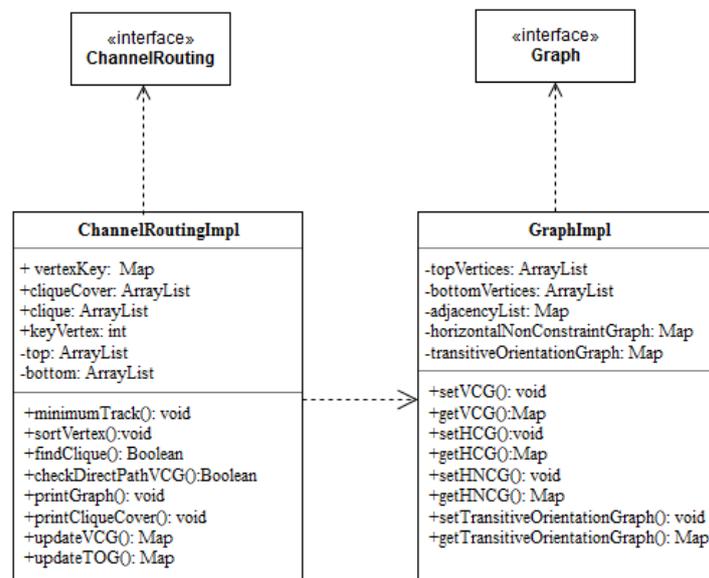


Fig.2. Class Diagram

The class ChannelRoutingImpl implements the abstract methods of ChannelRouting interface. Similarly the GraphImpl class implements the methods in interface Graph. ChannelRoutingImpl is dependent on GraphImpl to obtain the constraint graphs for further process of forming cliques and finding tracks for routing.

The methods implemented in GraphImpl mentioned below are used to generate constraint graphs.

setVCG(): It generates vertical constraint graph from the specification of the channel.

getVCG(): This gives the vertical constraint graph.

setHCG(): It generates horizontal constraint graph from the set of terminals specified.

getHCG(): This method gives the HCG.

setHNCG(): It generates horizontal non constraint graph from HCG.

getHNCG(): This method gives HNCG.

setTransitiveOrientationGraph(): It specifies the direction for the edges of HNCG.

getTransitiveOrientationGraph(): It helps to obtain the transitive orientation graph.

The methods in ChannelRoutingImpl class mentioned below are used to obtain clique cover and find tracks required for routing.

sortVertex(): It sorts the vertices in increasing order of their appearance on column.

minimumTrack(): It is used to obtain constraint graphs from GraphImpl and do further manipulations on them to obtain clique cover and find tracks for routing.

findClique(): It forms cliques considering the constraints.

checkDirectPathVCG(): This method checks if there is any direct path between two vertices in VCG.

updateVCG(): It merges the vertices which form clique to obtain updated VCG.

updateTOG(): The vertices which form clique and their corresponding edges are deleted to generate updated TOG.

printGraph(): It prints the constraint graphs.

printCliqueCover(): It prints clique cover, track count and tracks required for routing.

V. Results and Discussions

The algorithm implemented using JAVA generates the constraint graphs and finds the tracks required for routing. The set of terminals form the input for the routing problem. The figure below shows the input data.

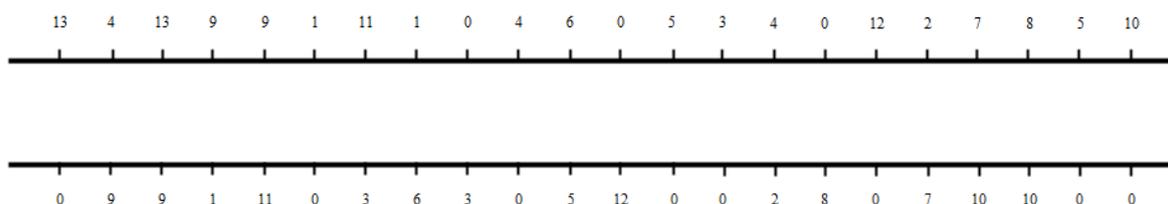


Fig.3. Input Channel

The vertices are sorted in ascending order according to their appearance on column. The VCG, HCG, HNCG and TOG are obtained and represented using adjacency list as shown below.

TABLE.1 VCG

Key	Value
4	9,2
13	9
9	1,11
11	3
1	6
6	5
2	7
7	10
8	10

TABLE.2 HCG

Key	Value
13	4,9
4	9,13,1,11,6,5,3,12,2
9	1,11,4,13
1	9,11,3,6
11	9,1,3

6	1,4,3,5
5	6,3,4,12,2,7,8,10
3	6,5,12,11,1,4
12	5,3,4,2,8
2	4,12,8,7
7	10,2
8	12,2,7,10
10	7,8,5

TABLE.3 HNCG

Key	Value
13	1,2,3,5,6,7,8,10,11,12
4	7,8,10
9	2,3,5,6,7,8,10,12
1	2,4,5,7,8,10,12,13
11	2,4,5,6,7,8,10,12,13
6	2,7,8,9,10,11,12,13
5	1,9,11,13
3	2,7,8,9,10,13
12	1,6,7,9,10,11,13
2	1,3,5,6,9,10,11,13
7	1,3,4,5,6,8,9,11,12,13
8	1,3,4,5,6,8,9,11,12,13
10	1,2,3,4,6,9,11,12,13

TABLE.4 TOG

Key	Value
13	1,2,3,5,6,7,8,10,11,12
4	7,8,10
9	2,3,5,6,7,8,10,12
1	2,5,7,8,10,12
11	2,5,6,7,8,10,12
6	2,7,8,10,12
5	Null
3	2,7,8,10
12	7,10
2	10
7	Null
8	Null
10	Null

The routing solution using four layers after forming cliques from constraint graph is shown below and colour description used for layers is given by Table 5.

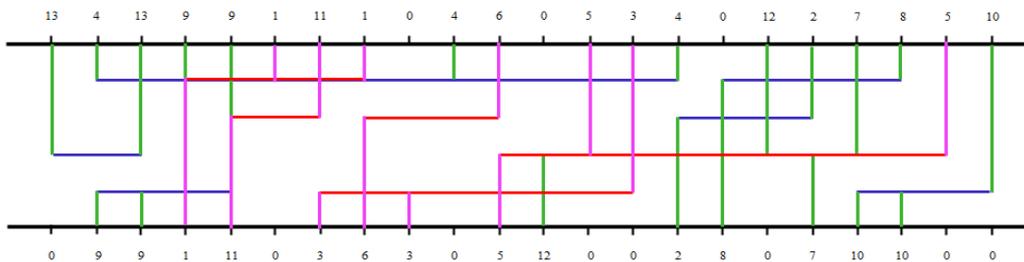


Fig.4. Routing Solution using Four Layers

The routing solution using two layers is given below and colour description for layers is given by Table

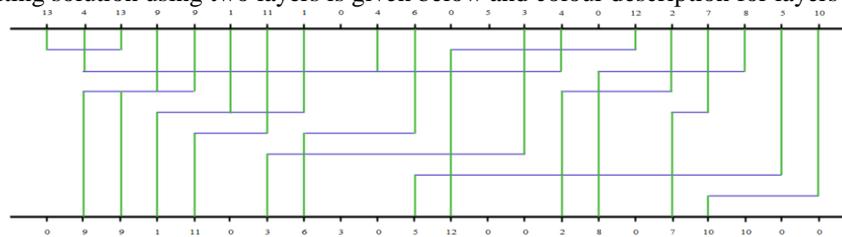


Fig.5. Routing Solution using Two Layers

TABLE.5 Colour Description of Layers

Colour	Description
Blue	First Horizontal Layer
Green	First Vertical Layer
Red	Second Horizontal Layer
Pink	Second Vertical Layer

The solutions for routing shown in figure 4 and figure 5 shows that the algorithm implemented using four layers minimize tracks for routing when compared to routing using two layers.

VI. Conclusion

The solution to channel routing problem is obtained by constructing the constraint graphs from channel specification. The cliques are obtained while considering the constraints which are represented using graphs. The algorithm is implemented using java which finds interconnection between nets. It gives fewer tracks for routing the channel using two horizontal and two vertical layers in comparison with one horizontal and one vertical layer for placing horizontal and vertical segments respectively. The reduction of tracks used for routing consequently reduces the integrated circuit area. The cyclic vertical constraints need to be handled in future. The user interface needs to be developed in the future to increase the usability.

References

- [1]. Sherwani N. A., "Algorithms for VLSI Physical DesignAutomation," Springer Private Limited, New Delhi, 1999.
- [2]. A. Pal, T. N. Maldal, S. SahaSau, 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.
- [3]. 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.
- [4]. 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.
- [5]. R. K. Pal, "Multi-Layer Channel Routing: Complexity and Algorithms," Narosa Publishing House, New Delhi, 2000.
- [6]. 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.