Noise Reduction in VLSI Circuits using Modified GA Based Graph Coloring

Timir Maitra[1], Anindya J Pal[1], Debnath Bhattacharyya[2], and Tai-hoon Kim[2,*]
[1]*Heritage Institute of Technology*
Kolkata 700107, India
timirmaitra@gmail.com, anindyajp@yahoo.com
[2]*Department of Multimedia*
Hannam University
Daejeon, Korea
debnathb@gmail.com, taihoonn@empal.com

Abstract

Analyzing and evaluating various noise avoidance techniques such as driver sizing, wire sizing, wire spacing and layer assignment. This paper presents an approach to solve the VLSI (very large scale integration) layer assignment problems that will leads to minimization of crosstalk noise. Our approach is based on Genetic Algorithm (GA) with introduction of modified GA operator, which is used to solve our problem that is formulated in a graph-coloring problem. In this paper, we study the problem of crosstalk minimization in 3-layer HVH channel routing without considering dogleg.

Key Words: *Channel Routing Problem, Layer Assignment, Crosstalk Noise, Coupling Capacitance, Horizontal and Vertical Constraint, Genetic Algorithm, Graph Coloring*

1. Introduction

Horizontal and Vertical wire segments are assigned in respective layers for interconnecting the nets in VLSI designing. Coupling capacitance between neighboring nets is a dominant component in today's deep submicron designs as taller and narrower lines are being laid out closer to each other. Considering the case of long overlapping wire segments on adjacent layers, there is a possibility of signal interference that may cause electrical hazards; termed as Crosstalk noise. The net on which noise is being induced is called the victim net whereas the net that induces this noise is called the aggressor net. Crosstalk noise affects a circuit in at least three ways: 1. inducing undesired signals in noisy environment, 2. changing the signal delay due to a switching transition and 3. increasing the switching power dissipation. Crosstalk depends on the routing of the lines, and hence the routing algorithms take this into consideration. Crosstalk distortion between two adjacent nets is determined by several factors like coupling capacitance between the nets, driving capacity of the nets, and timing of the signals.

Two adjacent wires form a coupling capacitor. In fact, the ratio of coupling capacitance is reported to be even as high as 70% ~ 80% of the total wiring capacitance, even in 0.25□m technology.

*Corresponding Author

The coupling capacitance between a pair of parallel wires is proportional to their coupling length, and is inversely proportional to their separating distance. The coupling capacitance between parts of orthogonal wires is negligible in comparison with the coupling capacitance between a pair of parallel wires for current technology. Given two wire segments i and j, the coupling capacitance between them, can be expressed as follows [2]:

$$c_c(i,j)=\frac{f_{ij}.l_{ij}}{d_{ij}}\frac{1}{\frac{1-w_i+w_j}{2d_{ij}}}$$

where w_i and w_j are the sizes of wires i and j ($w_i,w_j > 0$),f_{ij} is the unit length fringing capacitance between wires i and j, l_{ij} is the overlap length of wires i and j and d_{ij} is the distance from the center line of wire i to the center of wire j.

We are trying to minimize the coupling. During routing, we can control l_{ij}, d_{ij}, w_i and w_j. By avoiding overlap between two wires using optimized layer assignment.

A routing solution will not fail due to coupled noise if for all nets i [3]

$$\sum_{j\in A(i)}\left(\frac{R_j.C_j}{R_{io}.X_{ij}}+\frac{C_i}{X_{ij}}\right)^{-1}\leq \Xi_i\left(\sum_{j\in A(i)}X_{ij}\right)$$

where

$$C_i=C_g.l_i+\sum_{j\in A(i)}X_{ij}$$

Let introduce the notation used in the above sufficient condition for any grouting solution to be free of coupled noise problems. A(i) is the set of aggressor nets for net i and comprises all nets routed adjacent to it. X_{ij} is the coupling capacitance between nets i and j, C_g is the line-to-ground capacitance per unit length, l_i is the length of net i, R_{io} is the output resistance of the driver for net i, R_i is the driver resistance for net i, expression, C_i is the sum of the line-to-ground capacitance and all the line-to-line couplings from lines adjacent to line i and Ξ_i is the maximum tolerable amplitude function.

We find different previous work related to minimized crosstalk routing in [17] to [20]. The Crosstalk minimization problem in Channel Routing is modeled as a Layer Constraint Graph (LCG), where each node represents a net and two nodes in the LCG have an edge between them if corresponding net segments of same orientation (horizontally in same layer) share at least one tile in the routing grid. The channel routing problem is NP –complete [6] and therefore, there is no known deterministic algorithm to solve it in a polynomial time. New approaches are necessary to solve this problem.

Figure (1) shows an example of assigning wire segments *a, b, c, d, e* and *f* to the tracks of a horizontal panel with capacity K_p=4, and the constraint graph for the track assignment example is given.

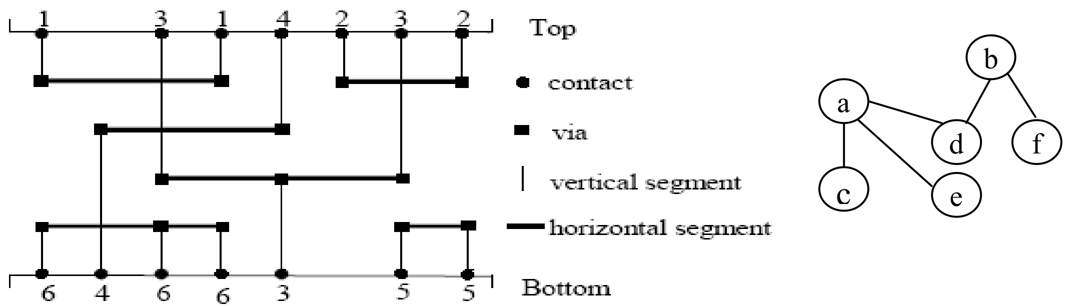

Figure 1. A channel layout and its constraint graph

Let us proceed to the formal statement of the problem. A channel is described by two sequences, *Top* and *Bottom,* in which the top and the bottom rows of the channel are placed respectively. The size of the both sequences is C (the number of columns in the channel). The set of nets is defined as *Net = {N1,.....,Nn},* where *n* is the number of nets. E.g. *Top= {1,0,3,1,4,2,3,2}; Bottom = {6,4,6,6,3,0,5,5},* where *C=8, n=6, Net={1,2,3,4,5,6},* the element 0 in *Top* or in *Bottom* denotes empty contact. The layout of the channel with connected nets is shown in Fig. 1.

The proposed approach to crosstalk minimization problem is based on few assumptions: (i) Vertical constraints are ignored partially, means we consider this constraint during track assignment but not at the time of crosstalk calculation, (ii) Number of terminal in a net may vary, (iii) Total number of tracks is not fixed. Consequently, it is reasonable to assume that there is crosstalk only between adjacent parallel wires because the horizontal capacitance can be several times larger than the vertical capacitance. Here we applied the solution of graph coloring problem to solve this. We find different previous work related to Graph Coloring in [6] to [15]. To solve the classical graph coloring problem we have used soft computing technique (Hybrid Modified GA).

2. Graph coloring solutions using soft computing and crosstalk minimization

In our approach we must identify the nets from the channel routing problem input that is upper boundary and lower boundary of the channel. Then consider every net as individual node of the graph. Connect the nodes through edges that will satisfy the constraints need to be taken care of. Thus we create a graph that represents the problem with constraint. When we are considering only Horizontal constraint at the time of plotting the edges, the resulting graph is called Horizontal Constraint Graph (HCG). It is a non-directed graph. Now we will consider vertical constraint to make a Vertical Constraint Graph, which is directed graph. Consider the effect of a directed path in the vertical constraint graph on the channel height. If doglegs are not allowed then the length of the longest path in VCG forms a lower bound on the channel height in the grid based model.

List of Constraints Considered

- Nets having horizontal constraint cannot be placed in same track.
- Nets having vertical constraint cannot be placed in same track.
- Nets having vertical constraints should be placed in higher track than the other, according to the order (direction) in VCG.

Vertical constraint graph must be acyclic. Then we merge HCG and VCG to form a graph that represents all constraints. In brief our algorithm has four modules as listed below:

a. Creating vertices and edges of a graph from the input considering the constraints

b. Finding out the coloration of the created graph with the specified number of colors using soft computing approach

c. Select the best coloration based on crosstalk with minimum number of tracks required

d. Preparing the final track assignment of nets from the above coloration

a. Creating vertices and edges of a graph from the list of constraints: A net Ni, in a grid based model, has a vertical constraint with net Nj if there exists a column such that the top terminal of the column belongs to Ji and the bottom terminal belongs to Jj and i != j. Vertical constraints are described by the directed Vertical Constraints Graph Gv = (ENet EV), where ENet is the set of vertices, corresponding to the set of the nets and EV is the set of links. Edge (m,n) in E exists if and only if net n is located above net m to prevent crossings of the vertical segments of nets. We represent this constraint using VCG, a directed graph. Algorithm for creating graph from the input is given below:

1. Identify nets
2. Find the overlap portion among nets to find Horizontal constraints
3. Create an adjacency matrix for HCG
4. Identify vertical constraints among nets
5. Create an adjacency matrix for VCG
6. Put an order to every nets that satisfy Vertical constraint
7. Merge HCG and VCG to make a single graph to operate on

b. Coloration of the created graph using soft computing approach: Here we have considered the GA with some modifications, as evolutionary algorithm. This section describes the main components of the Genetic algorithm for graph coloring. Integer strings encode the chromosomes.

This module starts taking the output of previous module that construct the constraint graph representing all the constraints.

i. Representation and fitness: In our algorithm, we represent the chromosomes as a set of integers (1,2,3….n). These integers are nothing but the color of the nodes. The positions of these integers are the node numbers for which those particular colors have been assigned. An example of a chromosome for a graph of 5 vertices is shown below:

1 4 3 2 1

Here the nodes 1 and 5 have the same color 1. 2^{nd} , 3^{rd} and 4^{th} nodes have the colors 4, 3 and 2 respectively. The fitness function is nothing but the total number of colors used, i.e., distinct integers in the chromosome.

ii. Initial population: The algorithm first builds an initial population, by randomly assigning colors to different nodes. In this pool, some chromosomes will denote valid coloration and some of them are invalid. But we have not discarded this invalid coloration. Instead, we use a repair operator rep_op, which coverts the invalid chromosome to a valid one. This rep_op checks whether two adjacent nodes have the same color or not. In that particular case, it replaces one such color by any randomly generated color.

iii. Special mutation: We have applied a special mutation operator to improve the chromosomes of the pool for a fixed number of iterations. Finally, the improved chromosomes are considered for the next generation. This process repeats for a prefixed number of iterations.

Sp_mutation

Step 1) Choose a particular chromosome

Step 2) Reduce the number of colors in that chromosome by one (by replacing one of the used colors by another used color)

Step 3) If the coloration is invalid then apply the *rep_ op*

(Here it'll try to repair the invalid coloration to valid coloration using the reduced number of colors available)

Step 4) If it succeeds or the coloration is valid

then place this chromosome in the pool

Else

goto step 1)

iv. Algorithm for Graph Coloring:

Steps of the Algorithm:

INPUT : The adjacency matrix of the graph created by the algorithm construct_graph i.

1. Create random initial pool of chromosomes (population)
2. Call fitness( ) to calculate fitness of each individual
3. Select best individuals based on fitness
4. Apply Reproduction on invalid chromosomes
   - 4.1 Generate random value p[0,1]
   - 4.2 If the p <0.5 then
     - 4.2.1 Crossover( )
     - 4.2.2 If new individual is fit then send to New population
     - 4.2.3 Send the new individual to new population
   - 4.3 Else
     - 4.3.1 Select some best population based on fitness value
     - 4.3.2 Call Special Mutation( ) for selected individual …
5. Calculate fitness of new population generated from Mutation
6. Insert best offspring in population
7. Evolution of stopping criteria

c. Preparing the final Channel Routing for Crosstalk Minimization from the above coloration: Once the above algorithm has generated the coloration, the next objective is to finalize the track assignment from the coloration using the following algorithm:

1. Calculate crosstalk for each coloration (chromosome in population)

2. Select best one with lowest crosstalk with lowest no of tracks needed to allocate the nets
3. Allocate track satisfying the order assigned to the nets in previous module
4. Optimized layer assignment with minimal crosstalk achieved

3. Computational experiments

In this section, we present experimental results in Table (1) obtained after executing the program on each graph 150 times as this number of execution gives near optimized result in average. The fundamental difference that makes this algorithm better than the normal Genetic Algorithm is use of Modified Mutation and Special Mutation instead of Normal Mutation in proper way. After implementation of these operators we found a abrupt change in result and it is positive. Here, the crossover operator remains same as the standard genetic algorithm's blind crossover operator. In this section we present the results of our algorithm on some benchmark graphs given at different papers included in reference. It has been implemented in C on Linux operating system. We have considered the population size as 10. We have considered the crossover probability (p) <=0.5. The no_of_iteration for *Sp_mutation* was set to 150 after trial and error. The value of max_iteration depends on the density and number of nodes of the graph.

Table 1. Results applying our algorithm on some benchmark Channel Routing Problem (after executing each graph 150 times)

Example	Dmax	Vmax	Minimum Track	My Minimum Track	Crosstalk	Optimized Crosstalk	No. of Track
RKPC1	3	3	4	4	15	1	7
RKPC2	3	4	6	6	11	11	6
RKPC3	4	3	5	5	11	7	6
RKPC4	4	4	5	7	9	8	7
RKPC5	4	4	6	6	12	10	7
RKPC6	4	5	7	9	18	4	13
RKPC7	5	3	7	8	17	17	8
RKPC8	5	5	7	9	15	13	10
RKPC9	6	6	10	12	40	11	18
RKEX1	10	7	-	17	73	46	21
RKEX2	14	4	-	25	118	86	30
RKEX3B1	12	12	-	22	126	43	40
RKEX3C	14	5	-	38	101	37	53
RKEX3C1	14	6	-	35	98	74	52
RKEX4B	17	13	-	38	361	224	55
RKEXPG	12	7	-	17	94	76	21
RANDOM1	19	6	-	39	380	112	73
RANDOM2	14	5	-	33	181	70	19
RANDOM3	16	7	-	36	240	60	74

Example	Dmax	Vmax	Minimum Track	My Minimum Track	Crosstalk	Optimized Crosstalk	No. of Track
EX A	10	7	-	6	28	28	6
EX B	6	3	-	9	14	4	13
EX C	9	13	-	13	67	67	13
EX D	5	3	-	6	22	22	6
EX E	5	4	-	6	16	1	9
EX F	4	3	-	4	9	3	6
EX G	5	3	-	8	24	2	12
EXYK1	10	7	-	17	70	53	20

4. Conclusions

We have presented experimental results on some of the benchmark Channel Routing Problem. We can not produce result compared with other algorithm's performance on those Channel Routing Problems are not available. The main purpose of this paper is to study genetic algorithm on graph coloring problem with new operator. In the future, we intend to refine this algorithm and apply it to color large complex graphs in reasonable time.

5. References

[1] Murat R. Becer, David Blaauw, Vladimir Zolotov, Rajendran Panda and Ibrahim N. Haji, "Analysis of Noise Avoidance Techniques in DSM Interconnects using a Complete Crosstalk Noise Model," *IEEE Computer Society*, 1999.

[2] A. Vittal and M.Marck-Sadowska, "Crosstalk Reduction for VLSI", *IEEE Trans. Computer-Aided Design,* Vol. 16, No. 3, pp.290-298, Mar. 1997.

[3] H. Zhou and D. F. Wong, "Global routing with crosstalk constraints," *IEEE TCAD*, 18(11):1683–1688, 1999.

[4] H.-P. Tseng, L. Sheffer, and C. Sechen, "Timing- and Crosstalk- Driven Area Routing," *IEEE TCAD*, 20(4):528–544, April 2001.

[5] T.G. Szymanski, "Dogleg Channel Routing is NP-complete", *IEEE Trans. on Computer-Aided Design*, Vol. 4, No. 1, pp. 31-41, Jan. 1985.Aided Design, Vol. 4, No. 1, pp. 31-41, Jan. 1985.

[6] D.C. Wood, "A technique for coloring a graph applicable to large scale time-tabling problems," *Computer Journal*, vol. 12, pp. 317-319, 1969.

[7] F. T. Leighton, " A graph coloring algorithm for large scheduling problems," *Journal of Research of the National Bureau of Standards*, vol. 84, no. 6, pp. 489-505, 1979.

[8] F.C. Chow and J.L. Hennessy, "Register allocation by priority based coloring", *Proceedings of the ACM SIGPLAN 84 symposium on Compiler Construction* Newyork, pp. 222-232, 1984.

[9] M.R. Garey, D.S. Johnson and H.C. So., "An application of graph coloring to printed circuit testing," *IEEE Transactions on circuits and systems*, vol. 23, pp. 591-599, 1976.

[10] A. Gamst, "Some lower bounds for class of frequency assignment problems," *IEEE Transactions on Vehicular Technology*, vol. 35, no. 1, pp. 8-14, 1986.

[11] Micheli G.D., BOOK TITLE *SYNTHESIS AND OPTIMIZATION OF DIGITAL CIRCUITS.* McGraw-Hill, 1994.

[12] S.S. Sarma, R. Mondal and A. Seth, "Some sequential graph coloring algorithms for restricted channel routing," *INT. J. Electronics*, vol. 77, no. 1, pp. 81-93, 1985.

[13] T.F. Coleman and J.J. More, "Estimation of sparse Jacobian Matrices and graph coloring problems," *SIAM. J. Numer. Anal*., vol. 20, pp 187-209, 1983.

[14] Garey, M.R. and D. S. Johnson, *BOOK TITLE COMPUTERS AND INTRACTABILITY : A GUIDE TO THE THEORY OF NP-COMPLETENESS*. Newyork, W. H. Freeman and Co., 1979.

[15] Baase, S. and A.V. Gelder, *BOOK TITLE COMPUTER ALGORITHMS:INTRODUCTION TO DESIGN AND ANALYSIS*. Addison-Wesley, 1999.

[16] H. H. Chen and C. K. Wong, "Wiring and Crosstalk Avoidance in Multichip Module Design," *Proc. Custom Integrated Circuits Conf*.,1992, pp. 28.6.1 – 28.6.4.

[17] T.Gao and C.L. LIU, "Minimum crosstalk channel routing," *Proc. Int. Conf. Computer-Aided Design*.1993, pp. 692-696.

[18] S. Thakur, K. Y. Chao and D.F. Wong, "An Optimal Layer Assignment Algorithm for Minimizing Crosstalk for Three-layer VHV Channel Routing", *Proc. IEEE International symp. On Circuits & Systems*, pp. 207-210, 1995.

[19] Vladimir N.Davidenko, Victor M. Kureichik , Victor V. Miagkikh, " Genetic Algorithm for Restrictive Channel Routing Problem,".

[20] Jens Lienig and K. Thulasiraman, "A Genetic Algorithm for Channel Routing in VLSI Circuits," *Evolutionary Computation, MIT Press*, Vol. 1, No. 4, pp. 293-311,1994.

[21] M. R. Becer, D. Blaauw, I. N. Hajj, and R. Panda. "Early probabilistic noise estimation for capacitively coupled interconnects," *SLIP*, pages 77–83, 2002.

[22] Achira Pal, Arpan singha, Shekhar Ghosh and Rajat K. Pal, "crosstalk Minimization in Two-Layer Channel Routing," *Proc. IEEE TENCON*, 2002

[23] David E. Goldberg. BOOK TITLE "GENETIC ALGORITHMS IN SEARCH, OPTIMIZATION & MACHINE LEARNING", Reading : Addison Wesley, 1989.