A Shortest-Path-Search Algorithm with Symmetric Constraints for Analog Circuit Routing[1]

Changxu Du, Yici Cai, Xianlong Hong, Qiang Zhou
EDA Lab, Department of Tsinghua University, Beijing, 100084, China

Abstract In this paper, we introduced a new algorithm, which is used for analog circuit routing, to search for the shortest path on the grid graph. It is different from previous path searching algorithms, since it concerns the symmetric constraint, one of the most significant performance requirements of analog circuits. We formulated the problem as a programming problem, and then used a heuristic method to solve it. Herein are two new cost functions that are the core of our algorithm. Tested by hundreds of industrial cases, the experiment results show that the algorithm could not only finish routing symmetric nets, with a high rate of success, but guarantee the performance of analog circuits.

1 Introduction

With the increasing demand of analog VLSI in the industrial community, the need for analog EDA tools, especially the layout generator, is skyrocketing day and night. Indeed, in an analog layout generation system, the router is one of the most difficult integrants. However, unlike that of digital circuits, performance in the analog circuit becomes the most significant factor that should be paid much attention to, so that in the routing phase, the issue of how to finish routing all the nets with the lowest delegation of circuit performance has been brought into focus. In detail, the appearance of analog circuit performance is reflected by both geometric and electronic constrains in the router. In other words, there are two main goals of analog circuit routing: one is to implement connections between nets with the least sources, or areas, and the other is to avoid delegation of circuit performance, or to meet both kinds of constraints.

Based upon those two goals, an ideal routing algorithm has to not only connect pins needed simply, but also consider all the constraints. In fact, among all the constraints involved in the analog circuit, that of symmetry is the prevalent one. Why do a couple of nets need to be symmetric? It is also attributable to the performance demand, which requires two singles to be transmitted through the absolutely same paths, with their transmission time and delegation same. Therefore, nets transmitting the corresponding signals must have the same shape, assuring their parasitic parameters are equivalent.

It is very difficult and complicated. Moreover, the irregularity of analog cells, which lead to the routing area irregular as well, and some forbidden regions, which cannot set any nets for avoiding performance delegation, result in serious shortage of available routing resources and the competition for these resources among prospective nets.

However, extant routers, such as ILAC[1], LADIES[2], and ANAGRAM II[3], do not involve the solution for symmetrical constraint and other parasitic parameters. Besides, they just used Maze Algorithm introduced by Lee in 1961[4], and Line-Search Algorithm introduced by Heyns in 1980[5], both of which work well when routing digital circuits, but they do not adapt to the analog circuit routing problem very well.

Therefore, it is necessary to develop a new routing algorithm that focuses on the special constraints, the limitation and irregularity of routing area, and parasitic parameters related to the analog circuit performance. Recently, a signal driven analog router was introduced by us[6], which began to concern above constraints.

The paper is organized as follows. In section 2, we explain the computing model of our algorithm, and define some essential concepts and formulate the main problem that the new algorithm focuses on. In section 3, we put forward the description of our new algorithm. In section 4, we demonstrate the results of the algorithm,

[1]This work is supported by National Natural Science Foundation of China(NSFC) 60476014 , Hi-Tech Research & Development (863) Program of China No.2005AA1Z1230

0-7803-9210-8/05/$20.00 ©2005 IEEE

and Section 5 concludes the paper.

2 Problem Formulation

In the algorithm, the grid-based routing model [7] is selected, and definitions of Grid Graph, Net, and Symmetrical Net could be found in [6] and [7]

Actually, it is a classical problem of searching for the shortest path between two points in a graph, and many mature algorithms, such as Floyd Algorithm, Dijkstra Algorithm, and A* Algorithm, have successfully solve the problem that only pays attention to the distance between two nodes, without additional constraints.

Nonetheless, when routing an analog circuit, symmetric nets could not be ignored, so that traditional algorithms are no longer appropriate, or even disabled. In fact, here the problem can be formulated as a programming problem, which is subject to three constraints.

Problem Formulation: Given a Grid Graph $G_{m,n} =< V, E >$, and a couple of symmetric nets:

$Net_1 =< n_1, n_2, ..., n_k >$, $Net_2 =< n_1', n_2', ..., n_k' >$

The axis is either $x = x_0$ or $y = y_0$.

Then, routing the two nets could be formulated as:

Goal Function: $\min \sum_{i=1}^{k} |(n_i, n_{i+1})|$,

such that $n_i \in V$ (1)

$$\sum_{i=1}^{k} |(n_i, n_{i+1})| = \sum_{i=1}^{k} |(n_i', n_{i+1}')| \quad (2)$$

$$|x_{n_i} - x_0| = |x_0 - x_{n_{i+1}}|, \text{ or } |y_{n_i} - y_0| = |y_0 - y_{n_{i+1}}| \quad (3)$$

Where, the coordinates of n_i is (x_{n_i}, y_{n_i}), the coordinates of n_{i+1} is $(x_{n_{i+1}}, y_{n_{i+1}})$. In the formulation, (1) represents that the algorithm is based on Grid Graph, so none of the nets is free out of the grid; (2) implies the symmetric constraint that then wire length of two symmetric nets are the same; (3) describes the relationship of two symmetric nets relative to the axis.

3 Algorithm Description

3.1 Execution Process of the Algorithm

The algorithm is made up of two phases: the orientation decision phase and the path expansion phase. The former phase computes the related parameters and decides the expansion orientation, and the later expands the path from one grid point to the next.

In the orientation decision phase, first of all, symmetric nets are split to two-terminal symmetric sub-nets, since the algorithm can only search for the path between two points on the Grid Graph. Then, starting at the source point, the algorithm calculates the expansion cost for the current point, and decides the expansion orientation. In detail, suppose that the current point is X, whose coordinate is (x_1, y_1), and the axis is $y = y_0, y_0 > y_1$. According to the constraint (3) mentioned in Section 2.2, the coordinate of the symmetric point, X', is $(x_1, 2y_0 - y_1)$. The cost function here would calculate the expansion cost derived from X firstly, and then calculate the same items derived from X'. Thereby, the factors that would affect the routing result, from both symmetric nets, are considered, so that the symmetric characteristic could be guaranteed well.

At last, in the path expansion phase, both the symmetric sub-nets are expanded to the next point, until the next point is the destination. Here, symmetric nets expand the same length at a time, in order to make the requirement of symmetry assured. Since the length of each step is the same, the constraint (2) mentioned in Section 2.2, $\sum_{i=1}^{k} |(n_i, n_{i+1})| = \sum_{i=1}^{k} |(n_i', n_{i+1}')|$, is guaranteed. The iteration works until all the sub-nets of a pair of symmetric nets are routed. The process of routing is described above, but implicit in it are cost functions, the spirit of our algorithm.

3.2 Cost Functions in the Algorithm

In the orientation decision phase, there is a cost function used to get different costs of choosing distinctive expansion orientations. Here lays a heuristic strategy, which estimates the possible cost in the future as well as the fixed cost until now. For Net_i, the cost function could be defined as:

$$TotalCost(Net_i) = CurrentCost(n_i) + EstimatedCost(n_i)$$

Where, $EstimatedCost(n_i)$ stands for the cost stimulated from the current node to the target node. Obviously, $EstimatedCost(n_i)$ is an estimated value, which, in the algorithm, is computed by the Manhattan

Distance between the current node and the destination. It provides the algorithm some heuristic information, which helps the algorithm determine the expand orientation close to the "globally best" one.

Compared with $EstimatedCost(n_i)$, $CurrentCost(n_i)$ is more complicated and precise. It stands for the cost stimulated from the source node to the current node; that is, it could be defined as:

$$CurrentCost(n_i) = \sum_{s=2}^{i} StepCost(n_{s-1}, n_s) \quad (4)$$

Where, $StepCost(n_{s-1}, n_s)$ is the other cost function that is used to get the comprehensive cost of one-step expansion, which is used to guarantee the requirement of symmetry and the demand of parasitic parameters. This function is defined as:

$$\begin{aligned} StepCost(n_{s-1}, n_s) = {} & \alpha \Delta R_{s-1,s} \\ & + \beta \Delta C_{s-1,s} \\ & + \gamma OrientationCost_{s-1,s} \\ & + \sigma \Delta CrossTalk_{s-1,s} \\ & + \delta RipCost_{s-1,s}(RoutedSubNets) \end{aligned} \quad (5)$$

Where, $\Delta R_{s-1,s}$ stands for the increment of resistance, and $\Delta C_{s-1,s}$ stands for the increment of capacitance in one-step expansion. As is well known, the resistance of a mental wire is related to its length, width and quality. In the routing phase, when the circuit structure is determined, the wire width is unchangeable, too. And, all the metal used in the circuit is the same, so their quality is also a constant. Therefore, the resistance of the mental wire varies only when the wire length changes. When the net expands a grid, resistance of the net increases as well, and so does the symmetric counterpart. Besides, capacitance between two nets is similar to the resistance, and its increment could also be calculated by the increment of wire length.

$OrientalCost_{s-1,s}$ stands for the cost of choosing different expansion orientations. When computing this parameter, there are three orientations for choosing in the Grid Graph, and the cost is determined by whether the future point is taken by some obstacle, and by whether the expansion orientation is changed from the horizontal to the vertical, or from the vertical to the horizontal. At the same time, the symmetric counterpart of the expanding net is also considered, to ensure the feature of symmetry. Herein lays the difference between previous algorithms and our algorithm. That is, this algorithm does not simply copy the result of one net to the symmetrical counterpart, but concerns both routing environment in the decisive function, avoiding conflicts before the real wire is added.

In addition, $\Delta CrossTalk_{s-1,s}$ stands for the increment of crosstalk after the expansion, also considering both the net and its symmetric net. Crosstalk is a kind of noise derived from two parallel nets. When routing a group of symmetric nets, every pair of horizontal nets are parallel, but the vertical pairs are not. Thereby, the cost generated by adding a horizontal path and that generated by adding a vertical one is different. To guarantee the circuit performance, this factor must be involved in the algorithm.

$RipCost_{s-1,s}(RoutedSubNets)$ is used in the rip-up and reroute phase, and it represents the cost of ripping a routed sub-net. Admittedly, it only useful when the rip-up and reroute phase is needed. $\alpha, \beta, \gamma, \sigma, and \delta$ are factors to adjust the weight of each term, and all of them could be adjusted by customers.

With those cost functions, we could precisely determine the expansion orientation and then get the shortest path between two points on the Grid Graph. By the algorithm, both symmetric nets are routed simultaneously, and the condition of routing areas, including obstacles, and the related parasitic parameters are considered at the same time. Moreover, the heuristic strategy improves the precision of solutions, and it also benefits to avoid abusing the limited resources.

4 Experiment Results

After implementing the algorithm, we used more than one hundred industrial cases to test it. These cases came from some EDA companies, which employ these cases as their benchmark. The scales of test cases range from dozens of nets to hundreds of nets, all of which contain some symmetric nets. The routing system is tested under the platforms of UNIX, LINUX, and WINDOWS, so the generality is proved.

The result showed that the algorithm is high efficiency and most of the success ratio of routing symmetric nets is 100%, and the usage of resources is satisfying, for the success ratios of all nets are also close to 100%. Besides, the symmetric constraint is met, whatever obstacles are symmetrical or not. Moreover, all the symmetric nets routed by the algorithm have passed the DRC, ERC, and

LVS, and the usage of routing resources is satisfied. As a result, problems mentioned in Section 1.2 are all solved in our algorithm, and relevant factors are all integrated into the algorithm.

Table 1 shows the number of nets in some cases, the number of symmetric nets, the number of routed symmetric nets, the success ratio of routing symmetric nets, and the success ratio of routing all nets.

Besides, Figure.1 shows some layout of the symmetric nets get by our algorithm.

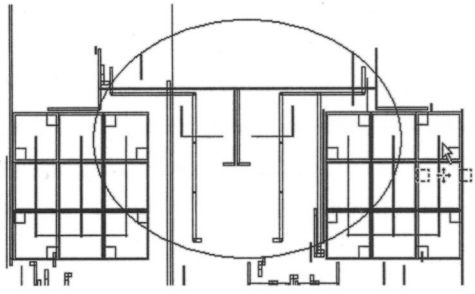

Figure.1. A pair of Symmetrical Nets in C4

Table 1 The Routing Results of Analog Circuits

Test Case	Total Number of Nets	Number of Symmetric Nets	Number of Routed Symmetric Nets	Success Ratio of Symmetric Nets	Success Ratio of All Nets
C1	85	3	3	100%	100%
C2	173	15	15	100%	100%
C3	230	5	5	100%	98.7%
C4	236	10	10	100%	99.2%
C5	249	5	5	100%	98.8%
C6	322	9	9	100%	98.4%
C7	323	7	7	100%	99.1%
C8	419	12	12	100%	98.8%

5 Conclusion

In this paper, we focus on a shortest path search algorithm with symmetric constraints, and implement the algorithm. This algorithm is designed for analog circuit routing particularly, since analog circuits have the specific performance requirements. By the algorithm, we could easily handle the difficulties in the analog circuit routing problem, and all the constraints are met so that the performance of analog circuits is guaranteed.

By the algorithm, we solve the problems of previous routers, including the difficulty of asymmetrical obstacles, the shortage of static algorithms, and the lack of consideration of parasitic parameters. It is clear that the experimental result is satisfying.

The experiment results shown above have proved that the algorithm is correct and efficient. Furthermore, we have expanded the algorithm to approach other kinds of nets in the analog IC router, including some nets required to be matched in parasitic parameters only.

6 References

[1] J.Rijmennants et al, "ILAC: An automated layout tool for analog CMOS circuits", IEEE J. Solid-State Circuits, (1989) 417-425

[2] Masato Mogaki, Naoki Kato, Youko Chikami, Naoyuki Yamada, Yasuhiro Kobayashi, "LADIES: An Automatic Layout system for Analog LSI's", ICCAD-89, (1989) 450-453

[3] J. M. Cohn, D. J. Garrod, R. A. Rutenbar and L. R. Carley, "KOAN/ANAGRAM Ⅱ : New Tools for Device-Level Analog Placement and Routing", IEEE Journal of solid State Circuits, Vol. 26, Issue 3, (1991) 330-342

[4] C. Y. Lee, "An Algorithm for path connections and its application", IRE Trans. On electronic computers, vol. EC-10, (1961) 346-365

[5] W. Heyns, W. Sansen, H. Beke, "A line-expansion algorithm for the general routing problem with a guaranteed solution," Proc. 17th ACM/IEEE Design Automation Conf., (1980) 143-249

[6] Jun Liang, "Classification Based IC Router", thesis for the master degree of Tsinghua University, (2004)

[7] K. Lampaert, G. Gielen,W. Sansen, "Analog Layout Generation For Performance and Manufacturability", Kluwer Academic Publishers, (1999)