C-024

A Fast Symmetrical Routing Algorithm based on Max-Flow Method

Zijiao Zhang†, Tieyuan Pan† and Takahiro Watanabe†

Abstract: In VLSI design, we often consider the routing for some special nets under the constraints such as length-matching and symmetry. The symmetrical routing is especially operated between the routing for the bus and the clock lines and the routing for most of the others without the priority. In this paper, the symmetrical routing algorithm based on Max-flow method is proposed for several nets routing in the multilayer. The algorithm solves the limitation of routing where layers and nets are less, to be applied in practical industrial products design. Moreover, the symmetrical rate is newly proposed for evaluating the rate of symmetrical proportion of length, bends and direction between corresponding nets in symmetrical routing. Experimental results show that the proposed algorithm has high symmetrical rate and efficiency and reduces the routing time.

1. INTRODUCTION

Routing is one of the most important processes in layout synthesis especially for modern analog and mix-signal circuit designs. It determines all the wire connections while minimizing the total wire length and satisfying all the user-specified routing constraints for better circuit performance. We often need to consider the routing of some special nets under the constraints such as equidistance and symmetry. The symmetrical routing is operated between the routing for the bus and the clock and the routing for most of the others without the priority. The symmetrical routing is the two nets routing in the special area.

To improve performance of an analog circuit, one of the major considerations is symmetry matching constraints for analog routing. Symmetry constraints are specified to route matched nets symmetrically with respect to some common axis. The symmetry constraint can reduce undesired electrical effects and balance the parasitic elements. Symmetrical routing region is divided into two parts: the symmetric regions and the intersection region. Ref.[2] proposed an algorithm to give a solution in the symmetric regions, where the left-side is solved by set-pair routing based on network flow and then the result is copied to the right-side. It makes symmetric regions completely symmetric, so our proposed algorithm is mainly to solve the problem of symmetry in the intersection region (Figure. 1).

For symmetry constraints, Ref.[1] proposed an algorithm to solve symmetrical routing which only uses the mixed integer programming. It takes lots of time and has low symmetrical rate. Later, [2] proposed an algorithm using mixed integer programming and set-pair routing which improve the efficiency and symmetric rate. However, when routing layout has intersections, this algorithm still takes lots of time and has low intersection region symmetrical rate. This paper provides an algorithm based on Max-flow method of the symmetrical

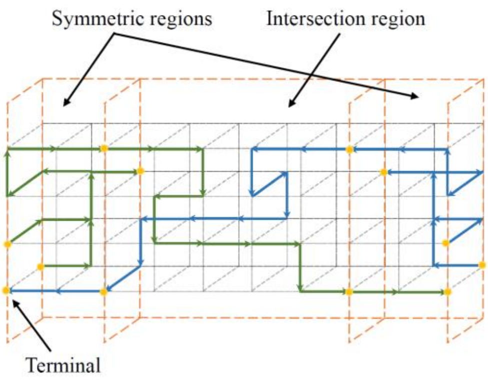

Fig. 1 Flow graph, symmetrical routing region and corresponding nets.

routing problem.

We consider the routing model with a grid graph in multilayer. The routing region is divided into the symmetric regions and the intersection region. All of pins are placed on the first layer. The symmetrical routing is converted into the problem of finding a Max-Flow for each layer and choosing the maximum symmetrical rate vertex. Additionally, the method can be modified to a new method of the symmetrical routing under the model where both horizontal segments and vertical segments can be routed in the identical layer.

2. PROBLEM FORMULATION AND OVERALL FLOW

The symmetrical routing is transformed into a problem based on a grid graph in this paper. Graph conceptions and symbols are defined as follows, similar to [2]-[4].

Given a graph G=(V,E), V= $\{v=(i,j,k)|1\leq i\leq X, 1\leq j\leq Y, 1\leq k\leq Z\}$ is a set of vertices where X, Y and Z are the maximum dimensions in x, y and z-axis direction, and

$E=\{e=(v,u)|v\in V,\ u\in\{(i-1,j,k),(i+1,j,k),(i,j-1,k),(i,j+1,k),(i,j,k-1),(i,j,k+1)\}$ is a set of edges where (v,u) is a directed edge.

Let $S=\{s_i|s_i\subseteq V, 0\leq i\leq m, k=0\}$ be a set of start terminals and $T=\{t_i|t_i\subseteq V, 0\leq i\leq m, k=0\}$ a set of end terminals, where m is the number of nets and k=0 since both the start and end terminals are in the first layer.

Let $N=\{n_i=(s_i,s_j,t_i,t_j)|s_i,s_j\subseteq S, t_i,t_j\subseteq t,\}$ be a set of the corresponding nets, O= $\{o_i|o_i\subseteq V, 0\leq i\leq q\}$ a set of the obstacles where q is the number of obstacles, and

$R=\{r_i=(s_i,t_i,p)|s_i\subseteq S, t_i\subseteq t, p\subseteq V, 0\leq i\leq m\}$ a set of routed nets where p is the vertices in the path. $r_i\cap r_j=\emptyset, 0\leq i,j\leq m, i\neq j$.

Input:

① The first layer of grid graph $G_1(V_1,E_1)$.

② The set of the start vertices S and end vertices T of the corresponding nets N and the obstacles set O.

Constraints:

----

† Graduate School of IPS, WASEDA University

① All nets should be routed from S to T simultaneously.
② The difference of the corresponding nets length should be minimized.
③ The routing nets not only can't intersect with each other but also avoid obstacles.

Output:

① The set of routing nets R.
② The symmetrical rate of each net and total symmetrical rate.

Objectives:

① Minimizing the total length of all nets. $L = \sum_{r_i \in R} |r_i|$
② Maximizing the symmetrical rate. (definition and description in section 3)

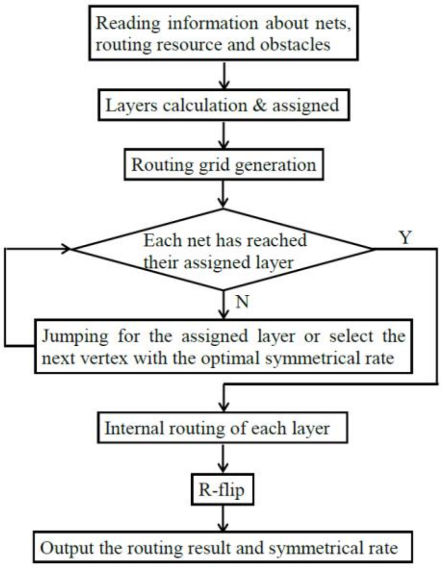

Fig. 2 Flow of the proposed algorithm.

The overall flow of our symmetrical routing algorithm is shown in Figure 2. First, the routing information is loaded, including the routing area G_1, the design rules, the routing obstacles O and the nets N to be routed. Second, nets are grouped. Then each group will be assigned to one layer. Third, when we get the total number of layers, the algorithm can generate the routing grid G. In the fourth step, the start vertices and the end vertices of each group travel to their assigned layer simultaneously. Next, each net is connected in their assigned layer. The final step is to reduce the length of the net by using R-flip [4].

In routing designs, both the runtime and symmetrical rate are very important for efficiency and quality. Therefore, the presented algorithm almost guarantees a better feasible solution for a given nets with max-flow method and prioritized consideration of symmetrical rate.

3. SYMMETRICAL RATE FUNCTION

We propose a new symmetrical rate function to obtain the symmetrical rate of feasible routes depending on corresponding segment's direction. In this algorithm, the next vertex of path selection depends on the symmetrical rate of feasible routes. Since symmetrical rate is a relative definition of the parameters by ourselves, and it cannot absolutely say that symmetrical rate can evaluate symmetry of routing result. We used the additional scientific data to compare the final result of our proposed algorithm and previous performance.

The total symmetrical rate function of routing nets is as follow.

$$sym = \frac{\sum_i c_i * l_i}{L}$$

l_i: The length of edge
(In grid graph, the length of each edge is 1)
c_i: The symmetrical coefficient
L: The total length of routing

Here, we discuss two possibilities of c_i which are divided into the first segment of each corresponding net and others.

CASE A. The segment is the first segment of each corresponding net.

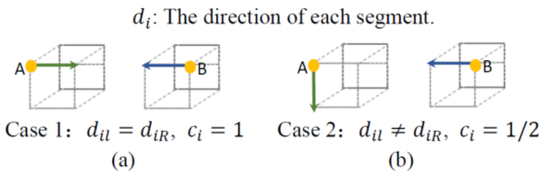

Fig.3 An example of CASE A.

Case A-1: The corresponding segments are in the corresponding directions. (Figure 3(a)) Since they are completely symmetrical, $c_i = 1$.

Case A-2: The corresponding segments are in different directions. (Figure 3(b)) $c_i = 1/2$.

CASE B. The segment is not the first segment of each corresponding net.

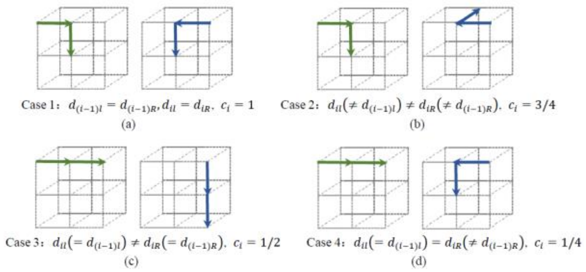

Fig.4 An example of CASE B.

Case B-1: Both of the previous segments and the current segment of corresponding nets are in the corresponding direction. Since they are completely symmetrical. $c_i = 1$. (Figure 4(a))

Case B-2: Both of the previous segment and the current segment of corresponding nets is in the different direction, and they all have bend. $c_i = 3/4$. (Figure 4(b))

Case B-3: Both of the previous segment and the current segment are in the same direction but the previous segments of corresponding nets are in different direction, which means there has no bend. $c_i = 1/2$. (Figure 4(c))

Case B-4: One of the previous segment and the current segment of corresponding nets is in the same direction and the other is in different direction, which means the one has bend and other one not. $c_i = 1/4$. (Figure 4(d))

With this function we can get the symmetrical rate of feasible routes in order to select a relatively high traverse.

4. SYMMETRICAL ROUTING ALGORITHM

In this section, we will describe the proposed algorithm step by step with an example in figure 5.

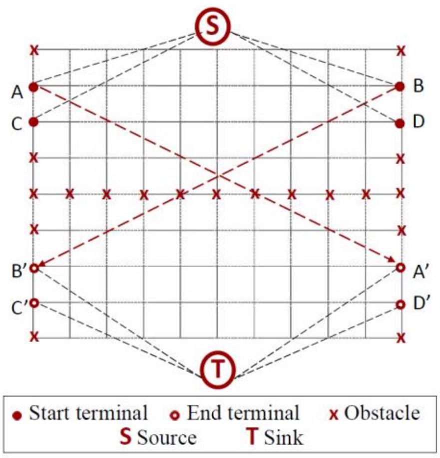

Fig.5 An example of the proposed algorithm.

4.1 Layers Calculation & Assigned

First, we discuss the feasibility of the max-flow method. Let S and T be the sets of the start and end terminals of the net N. The routing topology condition is defined in the following.

Max-Flow Method topology condition

- All terminals are put on the boundary of the routing area in the same layer.
- The boundary terminal sequence can be divided into the source terminal sequence S and the sink terminal sequence T, where T is the reverse of S and vice versa.

The boundary terminal sequence is $S = \{C, A, B, D\}$, $T = \{A', D', C', B'\}$ which violate the topology condition. At first, algorithm groups by finding the topology condition subsequence with the LCS algorithm. $S = \{S_1 = \{A, C\}, S_2 = \{B, D\}\}$, $\{T_1 = \{A', C'\}, T_2 = \{B', D'\}\}$. After grouping, each group will be assigned to one layer by calculate the max- flow. If a group doesn't satisfy the max-flow condition in a layer, it will be assigned to other layer of which max-flow is equal to the number of group nets. Here, S_1 and T_1 is assigned to layer 2, and S_2 and T_2 is assigned to layer 3.

4.2 Routing Grid Generation

In this step, the proposed algorithm determine the basic grid frame including the coordinate system and final adjacency link after obtaining the total number of layers. The coordinate system is used to calculating the Manhattan distance M_{ij} between two vertices and adjacency link is used by max-flow method and symmetric function.

4.3 Jumping for Each Net to the Assigned Layer

After the preparatory work is completed, the algorithm can start routing. This step is the most important step in the proposed algorithm which includes three constraints: the Manhattan distance between the current vertex and target vertex of corresponding nets M_{ct}, max-flow method and symmetrical function.

In this step, each time algorithm traverses all the feasible routes vertices which satisfy the max-flow condition and calculates their symmetrical rate. If the net is in the assigned layer, algorithm selects the next vertex with the optimal symmetrical rate. The other, it means the net will jump to the upper layer. Therefore, algorithm obtains the symmetrical rate of feasible routes vertices. If the upper layer's vertex has the optimal symmetrical rate, the net will jump to the upper layer. If not, under the circumstances that remaining length of the net is enough for the following route, the net does not jump to the upper layer but route in this layer and set a projection on the upper vertex to satisfy the max-flow method topology condition. This step is repeated until each net has reached their assigned layer.

4.4 Internal Routing of Each Layer

The algorithm will connect each net in their assigned

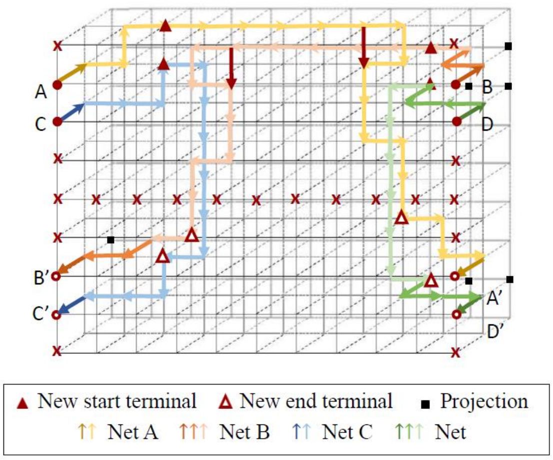

Fig.6 Result of the example.

layer after it has completed the jumping action. The next vertex of the path is still decided by the Manhattan distance s M_{ct}, max-flow method and the vertex with the optimal symmetrical rate. Then, we use R-flip to reduce the length of the net.

5. EXPERIMENTAL RESULTS

Table.1: Benchmark of Experimental Results.

benchmark	grid size	layers	nets	obstacle	intersections
data 1	6*4	1	1	0	0
data 2	160*300	1	19	0	0
data 3	5*5	2	2	0	1
data 4	11*9	3	4	19	3
data 5	15*10	4	10	27	8

Table.2: Experimental Result.

bench mark	mixed integer programming algorithm [1]				set-pair routing algorithm [2]				Proposed max-flow method			
	total length	time (s)	SR	LD	total length	time (s)	SR	LD	total length	time (s)	SR	LD
data 1	1	3062	100%	0	1	0.075	100%	0	1	0.069	100%	0
data 2	N/A	N/A	N/A	0	19	88.73	100%	0	19	37.07	100%	0
data 3	22	144.45	55.55 %	4	22	0.54	77.78 %	2	24	0.53	93.75%	0
data 4									68	3.99	94.12%	0
data 5									177	21.33	92.73%	0

- SR: Symmetrical rate LD: Length different between corresponding segments in exact matching.
- the time in [1] and [2] were re-calculated by conversion under the assumption of the same cpu used in our proposed method.

We implement our symmetrical routing algorithm in C programming, compiled by Xcode 7.2.1, and executed on a PC with 2.2GHz Intel Core i7 CPU and 16GB 1600MHz DDR3 RAM. Benchmark data 1~3 in Table 1 are same data used in [2] and data 4 and 5 are generated by ourselves. The terminals in all of data are put on the boundary of the routing area and the arrangement satisfies the max-flow method topology constraint. All obstacles are randomly generated. The data 1~2 don't have intersection wire in their intersection region, while the data 3~5 have some intersection wires. In Table 2, three algorithms are compared with respect to total path length, runtime, symmetrical rate SR proposed by ourselves and LD which is the difference between the length of the corresponding network with exact matching.

As shown in Table 2, the runtime of data 3 for our proposed method is larger than that of data 1 because of existence of intersection wires, even if its layout area is small. In all data, the difference between the lengths of the corresponding network with exact matching is 0, since the algorithm is based on Manhattan distance when it selects next vertex. The symmetrical rate shows that the proposed algorithm can obtain the routes of nets in the better degree of symmetry compared to the previous algorithm.

6. CONCLUSION

In this paper, we have proposed a fast symmetrical routing algorithm that satisfies symmetry matching constraints for analog nets. This research is aiming at finding shortest total length of routings and high symmetrical rate of routings. Due to considering the change of Manhattan distance in corresponding nets, the difference of corresponding nets length is minimized. Compared to the previous works, our algorithms explore the higher efficiency and better symmetrical rate by allowing extra bends in the matched routing results. A potential future research problem can be how to generate better routing grid to reduce the number of layers in some exceptional circumstances.

REFERENCE

[1] Akio Fujii, Takehiko Matsuo, Toru Fujimura, Bo Yang, Shigetoshi Nakatake, "MOS Analog Module Generation", IEICE Tech. Rep., vol. 107, no. 506 (2008).

[2] Masato Ito , Qing Dong , Shigetoshi Nakatake , "Symmetrical Routing based on Set-pair Routing and Mixed Integer Programming", IEICE Tech. Rep., vol. 114, no. 476 (2015).

[3] Xu Zhenghua , "The algorithm and application for the symmetrical routing", Transportation, Mechanical, and Electrical Engineering (TMEE), International Conference on, 16-18 (2011).

[4] Y. Kohira, A. Takahashi, " CAFE router：A Fast Connectivity Aware Multiple Nets Routing Algorithm for Routing Grid with Obstacles", 15th ASP-DAC, pp.281-286, 18-21, (2010).

[5] Hailong Yao, Yici Cai, Qiang Gao, "LEMAR: A Novel Length Matching Routing Algorithm for Analog and Mixed Signal Circuits", 17th ASP-DAC, pp. 157 – 162 (2012).

[6] Qiang Gao, Hailong Yao, Qiang Zhou and Yici Cai, "A Novel Detailed Routing Algorithm with Exact Matching Constraint for Analog and Mixed Signal Circuits", Department of Computer Science and Technology, Tsinghua University, P. R. China. Quality Electronic Design (ISQED), 12th International Symposium on (2011).