On Simultaneous Escape Routing of Length Matching Differential Signalings

Yen-Jung Lee, Hung-Ming Chen, and Ching-Yu Chin
Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan

***Abstract*—In substrate or PCB design, the escape routing problem is considered an essential part and has been widely studied in literature. There are industrial tools and some studies that work on simultaneous escape routing and escape routing of differential pairs on dense circuit boards. However, to route differential pairs simultaneously considering length-matching is still an important and on-going research problem. In this work, inspired by prior state-of-the-arts, we have implemented an integrated approach that achieves simultaneous escape routing considering length matching of differential pairs, our method avoids time-consuming ILP solutions in finding length-matching differential signal paths. Experimental results show that our approach can efficiently and effectively obtain length-matching of differential pairs on simultaneous escape routing to reduce differential-pair skews, compared with B-escape router we reimplemented.**

I. Introduction

As the scale of modern electronic systems increases rapidly, the design of printed circuit boards (PCB) becomes more and more complex. Escape routing problem is a key problem in PCB routing. "Escape" means that the pins on the components are trying to connect to each other, they try to escape from the inner side of components. The escape routing problem can be classified into ordered escape routing and unordered escape routing. The escaped wires around the grid boundary of the ordered escape routing are required to follow some ordering constraints while those of the unordered escape routing are not. In addition, ordered escape routing includes single-component escape routing problem and simultaneous 2-component escape routing problem. In this paper, we focus on simultaneous 2-component escape routing problem since it is a better way to obtain a real and complete solution [2].

Differential-pair routing [1] is widely used in high-speed PCB design. In each differential pair, one signal is transmitted by two complementary signals. These two signals are required to be transmitted in close proximity along a routing channel. Once the signal wires are routed close to each other, the noises on the channel can be simultaneously absorbed by the two signals. Aside from routing proximity, experimental results in [1] show that the length-matching differential pair obtains a smaller differential-pair skew than the non-length-matching one does. In fact, existing escape routing algorithms for simultaneous 2-component escape routing problem can handle differential pair routing problem [2], however it cannot guarantee that the two nets (from one pin to the grid boundary) of a differential pair are length matched. To further achieve length matching in the existing solution, extra routing resource and manually rerouting are required. Since a smaller differential-pair skew is one major factor to achieve better performance of differential pairs, we try to make two nets have similar wire lengths in this proposed work.

As for routing differential pairs, *Yan et. al.* [6] and *Li et. al.* [1] aim at the escape routing of differential pairs, while only [1] solves the length matching problem of differential pairs. Both of them focus on routing differential pairs for unordered single-component escape routing. As for 2-component simultaneous ordered escape routing, B-escape [2] is an algorithm with the best routability among recent researches. B-escape can accomplish differential pair routing, however, length matching of differential pairs is not taken into consideration. In addition, in the differential pair routing result of B-escape, once the two pins of the same differential pair in the same component are in different columns, the routing path of the differential pairs inside the same component may split. This could lead to worse differential-pair skew. In all, length matching of differential pairs is more meaningful in simultaneous 2-component escape routing, and it should be resolved.

Since industrial tools and previous works cannot solve the escape routing problem of differential pairs considering length matching, we propose an approach to work around. Our router is based on B-escape and can do net by net routing and differential pair routing at the same time. The differential pair length matching method is based on the concept of *min-cost median point* [1]. The wirelengths from the median point to the two pins of the differential pair are equal and the shortest, satisfying length matching of differential pairs. However, since we observe that [1] has problems in runtime and solution space limitation, we have made efforts to improve the approach. Experimental results show that our escape routing architecture guarantees that all differential pairs are length matched. Consequently, the differential-pair skew of each differential pair is minimum.

The rest of this paper is organized as follows. Section II describes the problem and flow, and Section III shows the basic ideas of B-escape, including boundary route and dynamic net ordering. Our implementation details are also mentioned in the same section. Section IV introduces our strategy to route differential pairs in two components considering length matching. Experimental results are shown in Section V followed by conclusions in Section VI.

II. Problem Formulation and Our Proposed Flow

The objective of simultaneous 2-component escape routing is to route all terminal pins in two components to the component boundaries with the same ordering. In order to satisfy the design specification including differential-pair skew, differential pairs should be designed carefully to maintain the performance. Length matching on differential pairs for simultaneous escape routing is described as follows.

Problem 1: $A1, A2$ are two given arrays of $p \times q$ and $r \times s$ pins with capacity c (that means no more than c nets are allowed to pass through any 2 neighboring pins), which is a user-defined parameter[1]. Given n differential pairs with pins $\{(P1_a, P1_b, P1_A, P1_B), ..., (Pn_a, Pn_b, Pn_A, Pn_B)\}$ and m single nets $\{N_1, ..., N_m\}$, the problem of simultaneous escape routing considering length matching of differential pairs is to find a routing path for each differential pair and single nets from their pins to the grid boundary in two components with the same net ordering, and all of the differential pairs are length matched.

When it comes to route differential pairs, our length matching approach includes two stages. First, we find a suitable min-cost median point and a proper path which connects two pins by shortest and equal wire lengths without crossing routed nets. Second, we escape route the median point to the component boundary and make an order check to guarantee the routing order of differential pairs in two components are the same. The routing process continues until a solution is found or the times of backtracking have exceeded the limit.

III. Implementation of Simultaneous Escape Routing

Existing published algorithms [3][4] for simultaneous escape routing problem are based on pattern routing. No more than two L-shaped fixed routes above/below the pin are given to each pin.

[1]The illustrations shown in this paper use c=2, the regular setup in previous works. Diagonal capacity constraint is not considered in this work.

978-1-4799-2311-3/13/$31.00 ©2013 IEEE

The routing space for each pin is very small. In this case, if the pins on the two components are aligned in similar orderings, these algorithms would perform well. However, more complicated escape problems cannot be solved by escape algorithms based on fixed/limited escape patterns.

Another published algorithm is B-Escape [2], which is a simultaneous 2-component escape routing algorithm based on boundary routing approach. B-Escape can solve complicated escape problems in short time. The algorithm was tested on a set of industrial escape problems, which were previously successfully routed by experienced layout experts taking about 8 hours per problem. B-Escape successfully solved all of the problems within minutes while Cadence Allegro PCB router was only able to complete the routing of half of the problems. Below we describe boundary routing and net ordering in details.

A. Boundary Routing

Boundary routing is the foundation of B-escape [2] algorithm. In order to present the idea of boundary routing, we use a rectangular routing area to represent component and assume that there is a grid structure inside it. In simultaneous 2-component escape routing problem, two components are face to face. The selected grid points to be routed are to escape to the three boundaries drawn in thick lines. To avoid net crossing in the inter-component area, the routing ordering along the 3 boundaries of two components (for the left component, the order from corner a to corner b clockwise, for the right component, the order from corner c to corner d counterclockwise) must be the same.

Observation shows that if a pin is routed following the routing boundary, more space will be available for later pins ([5] also presents the same idea). In this way, more pins will be able to escape route to the boundary. This leads to the boundary routing strategy: whenever we route a pin, we first route it to the routing boundary and then follow the boundary. After routing a pin, the routing boundary shrinks to exclude the routing path of that pin. In [2], it is proved that if a routing solution exists, it can be captured by the boundary routing strategy. If we change the way we route from the pin to the routing boundary and the direction we follow the boundary, different routing styles can be created. Six routing modes were found simple and effective in B-escape to get a boundary routing solution that meets the simultaneous escape routing problem: upward, downward, up-down, detour upward, detour downward, and detour up-down.

B. Dynamic Net Ordering

According to the complexity of escape situations on PCBs, finding the correct ordering for the 2 components beforehand is nearly impossible. Therefore, B-escape [2] uses a dynamic strategy to solve the ordering problem. The idea is to gradually determine the routing order as we route the nets. Whenever routing a new net, B-escape tentatively routes each remaining net, evaluates the routing cost and picks the one with minimum cost. Cost is calculated dynamically based on current routing status, consisting of two factors: one is the number of pins trapped (unroutable) by routing current net; the other is the number of pins blocked (still routable) by current net[2].

Though choosing the net with minimum cost at each step is a good idea, it cannot always guarantee the best routability, sometimes pins could be trapped by routed nets. Therefore, a reorder method is applied. For each step, nets/pairs are sorted by its routing cost in non-decreasing order. The net/pair with minimum cost will be chosen. Once an unrouted pin is trapped by routed nets, backtrack is performed to the step where the cost difference between the first net/pair and the next candidate is minimum. Then the next candidate net/pair will be chosen instead and the routing process goes on. By reordering, the router can make sure that all the pins remain routable at each routing step. B-Escape loops through six routing modes from upward mode to up-down detour mode and finally outputs the solution with the best routability.

[2]It is worth mentioning that for differential pair routing (not mentioned in [2]), no matter when we are doing B-escape non-length-matching routing or our length-matching routing, we route the 2 pins of a differential pair in both components and count its differential pair cost.

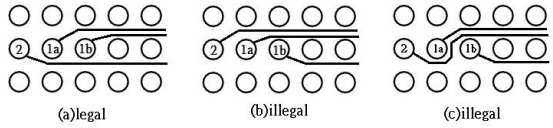

Fig. 1. Differential pair routing solutions. (a) shows legal solution and (b)(c) show illegal ones due to separation. Length matching is not considered here yet.

IV. Length Matching in Simultaneous Escape Routing

As mentioned in Section I, the two nets of a differential pair should be routed together. Fig. 1 shows three routing solutions, the circles represent the pin nodes in the component, where net $1a$ and net $1b$ are a differential pair. The first one (a) satisfies differential pair routing constraint. The second one (b) is illegal because the paired nets are separated by a row of pins. The third one (c) is also illegal because the paired nets are separated by another net. While the differential pair routing method in [2] did not solve length matching problem, a state-of-the-art work considering differential pair length matching problem in single-component escape routing [1] gives us the insight to resolve such issue: the concept of *min-cost median point* which connects two pins by shortest and equal wirelengths.

A. Finding Min-Cost Median Point

Different from the differential pair routing strategy used in B-escape, we route the two nets of one differential pair at the same time. Here we use the concept of min-cost median point mentioned in [1]. A min-cost median point for a differential pair is a median point which has the shortest and equal Manhattan distances from the median point to the two pins of the differential pair. Since the Manhattan distance between the median point and the two pins of the differential pair are equal, length matching of differential pairs can be achieved by two steps. First, we route from the two pins of a differential pair to a min-cost median point. Second, we route the two nets in a differential pair from the min-cost median point to the component boundary together. Note that for each differential pair, the min-cost median points lies on tile nodes, which are at the center of each four neighboring pins. Thus, differential pairs are no longer routed with grid structure with switch boxes in B-escape. Instead, differential pairs are routed through tile nodes to achieve length matching.

In order to find min-cost median points, we apply an efficient algorithm called MCMPF [1] (Min-Cost Median Point Finding) to find the min-cost median points for each differential pair. Once the position of the two pins of a differential pair is given, MCMPF algorithm can find all possible min-cost median points.

B. Shortest Pin-to-Pin Path through Median Point Enumerating

After all median points of the differential pair are found, since there may be more than one possible median points for one differential pair and more than one suitable paths for the chosen median point, the next step is to choose one median point and a valid pin-to-pin path for the differential pair. As shown in Fig. 2, we list four of eight possible pin-to-pin paths for $1a$ and $1b$ ((a), (b), (c), (d)). In (e), if net 2 is routed beforehand, routing the possible pin-to-pin paths (a) and (b) would cause crossings. Therefore, it is necessary to check the routability of each possible solution before routing them.

Now the problem is, among the possible median points and pin-to-pin paths, which solution is the best? When we are choosing a good median point and a valid pin-to-pin path, we sort the possible median points and pin-to-pin paths through the median points first and then sequentially check whether the pin-to-pin path will cross another routed net. If the first candidate chosen will cross another routed net, we check the next candidate and move on until a solution is found.

To improve routability, we sort the candidates of median points and pin-to-pin paths by *the space left for other unrouted nets* after routing it. Different median point and pin-to-pin path

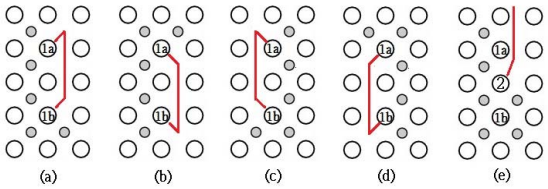

Fig. 2. Possible routing paths and possible crossing: four possible pin-to-pin paths (a)-(d). (e) shows future possible crossing.

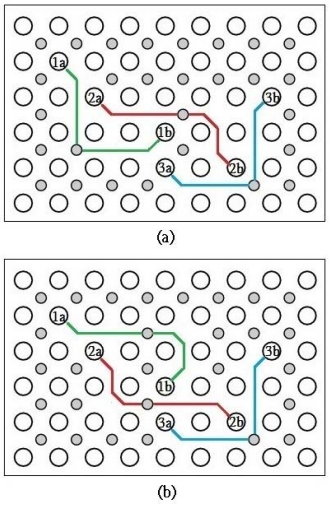

Fig. 3. Two pin-to-pin path solutions to 3 differential pairs, they have different set of possible routing orders. This shows the drawback of ILP method to determine routing paths.

leave different routing space for the unrouted pins. If a median point and a pin-to-pin path chosen could leave more routing space for the other nets than another one, then it is a better candidate. For example, when we are routing the left component with upward mode (Fig. 2), choosing tile b as median point leaves more routing space for other unrouted pins than choosing tile a, c, and d, tile b is then a better median point candidate. In this case, we first choose the median points with the largest y-axis, then the median point with the largest x-axis. After we have chosen a median point, there could be several possible pin-to-pin paths through the median point.

Note that as routing mode changes, the priority of the same routing candidate changes. For example, as shown in Fig. 2, (a) has a higher priority than (b) when routing up mode/up-detour mode, while (b) has a higher priority than (a) when routing down/down-detour mode, depending on the routing space left for unrouted nets after routing them. Here we do not employ the ILP method used in [1] to predetermine the pin-to-pin path to every differential pair for two reasons. First, ILP will spend much time to find solutions. Second, it will result in the limitation of the solutions when performing simultaneous routing. If we predetermine the pin-to-pin paths by ILP, some differential-pair-to-boundary path and routing orderings at the component boundary would be eliminated. For example, assuming 3 differential pairs lie in the left component, Fig. 3 shows two possible pin-to-pin path solutions prerouted by the ILP method used in [1]. Based on B-escape routing modes (shown in Table I), the possible routing orderings at the component boundary to Fig. 3(a) are (1,2,3), (2,3,1), and (3,1,2) while the possible routing orderings to Fig. 3(b) are (1,3,2), and (3,2,1). These show that the solutions from ILP method will constrain the routing results.

C. Median-Point-to-Boundary Path Determination Considering Net Ordering

After a pin-to-pin path through min-cost median point is found, the next step is to route from min-cost median point to the component boundary. Here the boundary routing strategy [2, 5] is applied on the selected median point, therefore a routing path from median point to the component boundary is obtained.

In this step, we should notice that when we are routing from min-cost median point to the component boundary, the differential pair net ordering must be taken into consideration; otherwise crossing may occur in the inter-component area. Fig. 4 shows the routing result without considering the net ordering of differential pairs, where $1a$ and $1b$, $1A$ and $1B$ are corresponding differential pairs in two components. In order to avoid crossings in the area between components, net ordering should be considered.

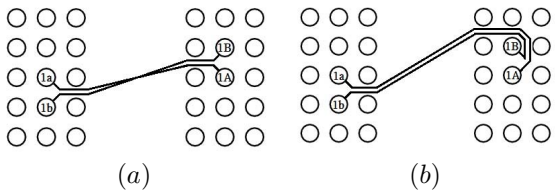

Fig. 4. (a)Incorrect ordering of differential pairs in two components causes crossing. (b)Routing direction "up" in the second(right) component will not cause crossing.

When we are routing differential pairs, we route the differential pair in the first component, then the second, therefore the net ordering of differential pairs is decided by the routing result in the first component. To match the net ordering of differential pairs in two components, the net ordering in the second component must meet that of the first component. By observation, we find that there are four directions, "up", "down", "left" and "right" for each median point to do tile-to-tile boundary route at the median-point-to-boundary routing stage. And the net ordering of differential pairs at the boundary are decided by the relative position of the pins of a differential pair and the direction the median point goes at the median-point-to-boundary routing stage. As a result, after the median point and pin-to-pin path are selected in the second component, when we are routing from the median point to the boundary, we need to check whether the direction the median point goes to will cause the same net ordering in the first component. If a proper routing direction is selected for the median point in the second component, no crossings will occur in the inter-component area.

For instance, if we route the left component first in Fig. 4(a), routing the median point in the right component to "left" would cause crossing. On the contrary, in Fig. 4(b), if we route the median point in the right component to "up", the ordering of the differential pair in the two components will be matched. In addition, if we always route the left component first, then the right component, crossings will still occur. Sometimes we cannot find a proper direction for the median point in the right component to meet the routing ordering of the first component.

As shown in Fig. 5(a), if we route the left component first, the median point in the right component can only go to the "left" direction, causing a crossing in inter-component area. In this case, we will try to route the right component first, the left component next and check whether the direction the median point in the left component goes to will get the same net ordering as the right component. As shown in Fig. 5(b), we route the differential pair in the right component first, the median point in the left component can be routed in a proper direction to meet the ordering of the differential pair in the right component.

Based on previous observation, we have developed an algorithm performing differential pair length matching. Each time we route a differential pair in simultaneous 2-component ordered escape routing based on B-escape, the algorithm is employed. We route the differential pair in the left component first and the right component next. If no path is found, we reverse the order in that differential pair.

V. Experimental Result

Our router is implemented in C++ and experiments are performed on a Intel Xeon E5620 CPU system with 70GB

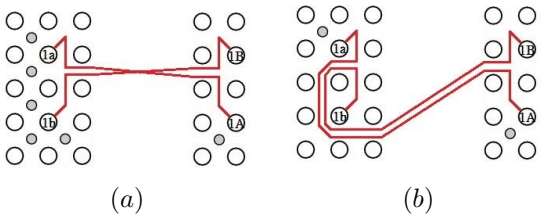

Fig. 5. (a)Inappropriate routing direction for the second(right) component. (b)Route the right component first can obtain feasible solution.

TABLE I
BENCHMARK INFOMATION

Case	Left Component #Row*#Column	Right Component #Row*#Column	#Diff.pair	#Single net	#Total net
1	10 × 10	10 × 10	5	0	10
2	14 × 14	14 × 14	10	0	20
3	14 × 14	14 × 14	12	0	24
4	20 × 20	20 × 20	20	0	40
5	10 × 10	10 × 10	3	4	10
6	14 × 14	14 × 14	4	10	18
7	14 × 14	14 × 14	5	12	22
8	20 × 20	20 × 20	10	20	40

memory. We test 8 cases where all differential pairs are generated according to the information of the distribution of the distances between the two pins of a differential pair from [6]. Our benchmark information is shown in Table I, where "#Diff. pair" gives the number of differential pairs, "#Row*#Col" gives the size of the left and right component. The first four cases are composed of differential pairs while the latter four cases contain both single nets and differential pair nets.

To better compare the routability, in non-length-matching B-escape router, when a pin at the component boundary performs escape routing, it occupies a track of routing capacity besides it. Additionally, in our implementation of non-length-matching B-escape router [2], given a differential pair 1a and 1b, if sequentially routing 1a and 1b will cause a failure while routing 1b first and 1a next is a success, the router will accept this solution and the routing process will continue without backtracking.

Table II shows our experimental results, where "Avg. len." gives the average wire length of a net in one component, and "Equal len. rate" gives the rate of length matching of differential pairs. They show that our work can achieve 100%

TABLE II
EXPERIMENTAL RESULTS

Case	Non-length-matching B-escape[2]				Our Work			
	Routability	Time(s)	Avg. len (tile)	Equal len. rate	Routability	Time(s)	Avg. len (tile)	Equal len. rate
1	100%	0.69	10.72	0%	100%	0.04	6.87	100%
2	100%	26.37	11.83	0%	100%	0.64	10.23	100%
3	100%	41.02	10.57	0%	100%	1.25	7.63	100%
4	96%	1678.45	–	–	100%	15.44	9.58	100%
5	100%	0.91	9.68	0%	100%	0.22	7.22	100%
6	94%	12.92	–	–	100%	7.82	10.11	100%
7	95%	31.14	–	–	100%	25.78	9.62	100%
8	96%	1350.92	–	–	100%	608.67	14.59	100%

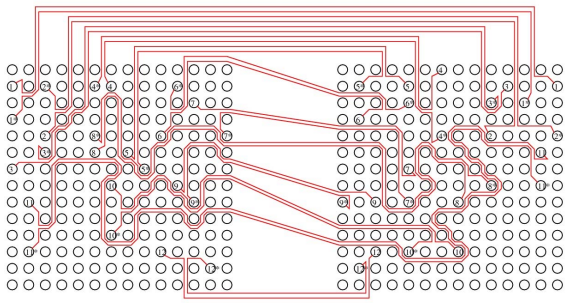

Fig. 6. Routing result of case 3 obtained by our work.

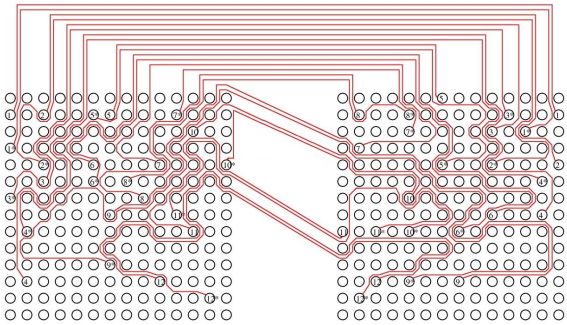

Fig. 7. Routing result of case 3 obtained by non-length-matching B-escape. Length is not matched in differential pairs.

routability in all test cases while B-escape can only completely route four of them. With less average wire length, we achieve 100% differential pair length matching in all cases, while no differential pairs are length matched in the result of B-escape. Because B-escape does not perform length matching of differential pairs, it is almost impossible for a differential pair in both components to obtain length matching result. By experiments in [1], the differential pair skews are well improved with our length matching result.

Our work greatly reduces the runtime than that of B-escape, especially in the first four all-differential-pair cases. There can be two reasons. First, each time non-length-matching B-escape routes a differential pair, the two nets of it are routed separately, and boundary routing are executed two times consecutively. In contrast, our work routes the two nets of a differential pair together at one time, therefore, boundary routing are only executed once. This is also the reason why routability and wirelength are better by our work. Second, due to the switch box structure, the non-length matching B-escape router has to check more possible routing directions in each routing stage. Fig. 6 and Fig. 7 show the differential pair routing result of case 3 obtained by our work and non-length-matching B-escape, ours achieves length-matching requirement.

VI. CONCLUSION

In this paper, based on the B-escape routing algorithm [2] and the idea of *min-cost median point* used in differential pair length-matching strategy [1], we create a work that achieves simultaneous escape routing considering length matching of differential pairs. On routing differential pairs, we first find a min-cost median point and a pin-to-pin path through the median point that guarantee equal wire length. Second, we apply boundary routing to the median point to get the path from median point to boundary. Experimental results show the effectiveness and efficiency of our approach.

REFERENCES

[1] T.-H. Li, W.-C. Chen, X.-T. Cai, and T.-C. Chen. Escape routing of differential pairs considering length matching. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 139 – 144, January 2012.
[2] L. Luo, T. Yan, Q. Ma, M. Wong, and T. Shibuya. B-escape: a simultaneous escape routing algorithm based on boundary routing. In *ACM International Symposium on Physical Design*, pages 19 –25, 2010.
[3] M. Ozdal and M. Wong. Simultaneous escape routing and layer assignment for dense pcbs. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 822 – 829, November 2004.
[4] M. Ozdal, M. Wong, and P. Honsinger. Simultaneous escape-routing algorithms for via minimization of high-speed boards. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(1):84 –95, January 2008.
[5] T.-Y. Tsai, R.-J. Lee, C.-Y. Chin, C.-Y. Kuan, H.-M. Chen, and Y. Kajitani. On routing fixed escaped boundary pins for high speed boards. In *IEEE/ACM Design, Automation, Test in Europe (DATE)*, pages 461–466, 2011.
[6] T. Yan, P.-C. Wu, Q. Ma, and M. Wong. On the escape routing of differential pairs. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 614 –620, November 2010.