

FXT-Route: Efficient High-Performance PCB Routing with Crosstalk Reduction Using Spiral Delay Lines

Meng Lian

Technical University of Munich
Munich, Germany
m.lian@tum.de

Yushen Zhang

Technical University of Munich
Munich, Germany
yushen.zhang@tum.de

Mengchu Li

Technical University of Munich
Munich, Germany
mengchu.li@tum.de

Tsun-Ming Tseng

Technical University of Munich
Munich, Germany
tsun-ming.tseng@tum.de

Ulf Schlichtmann

Technical University of Munich
Munich, Germany
ulf.schlichtmann@tum.de

ABSTRACT

In high-performance printed circuit boards (PCBs), adding serpentine delay lines is the most prevalent delay-matching technique to balance the delays of time-critical signals. Serpentine topology, however, can induce simultaneous accumulation of the crosstalk noise, resulting in erroneous logic gate triggering and speed-up effects. The state-of-the-art approach for crosstalk alleviation achieves waveform integrity by enlarging wire separation, resulting in an increased routing area. We introduce a method that adopts spiral delay lines for delay matching to mitigate the speed-up effect by spreading the crosstalk noise uniformly in time. Our method avoids possible routing congestion while achieving a high density of transmission lines. We implement our method by constructing a mixed-integer-linear programming (MILP) model for routing and a quadratic programming (QP) model for spiral synthesis. Experimental results demonstrate that our method requires, on average, 31% less routing area than the original design. In particular, compared to the state-of-the-art approach, our method can reduce the magnitude of the crosstalk noise by at least 69%.

CCS CONCEPTS

• Hardware → PCB design and layout; • Mathematics of computing → Integer programming.

KEYWORDS

PCB routing, crosstalk noise, spiral delay line, mixed-integer-linear programming, quadratic programming

ACM Reference Format:

Meng Lian, Yushen Zhang, Mengchu Li, Tsun-Ming Tseng, and Ulf Schlichtmann. 2023. FXT-Route: Efficient High-Performance PCB Routing with Crosstalk Reduction Using Spiral Delay Lines. In *Proceedings of the 2023 International Symposium on Physical Design (ISPD '23)*, March 26–29, 2023.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISPD '23, March 26–29, 2023, Virtual Event, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9978-4/23/03...\$15.00
<https://doi.org/10.1145/3569052.3571873>

Virtual Event, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3569052.3571873>

1 INTRODUCTION

As modern circuit clock frequency reaches the gigahertz range, the timing constraints become increasingly tight. Delay matching, therefore, has become a major consideration of modern printed circuit board (PCB) routers to improve the performance of designs [6, 13, 14]. As the propagation time of a signal is directly proportional to the wire length, adding *delay lines* to the shorter wires is the typical method to introduce a propagation delay between circuit board elements to synchronize the signal. To address the delay-matching issue, Mustafa Ozdal *et al.* allocate extra routing resources for length compensation using a Lagrangian relaxation framework [6]. Yan *et al.* transform the length-constrained routing problem into an area assignment problem and solve it with bounded-sliceline grids (BSG) [13, 14].

The methods mentioned above extend wires to sufficient lengths through meandering transmission lines, as shown in Figure 1 (a). This design, referred to as the *serpentine delay line*, is commonly used for delay matching. However, in high-speed digital circuit applications, serpentine delay lines introduce, besides the desired delay, spurious dispersion that makes the signal arrive sooner than expected. This phenomenon, known as the *speed-up effect*, is mainly attributable to the synchronous accumulation of the crosstalk noise between adjacent wire segments. Specifically, due to the periodicity of the serpentine structure, the crosstalk noise signals induced at different times may accumulate to trigger an erroneous logic gate [7, 11, 12].

Since crosstalk noise is inversely proportional to the distance between parallel wire segments, Tseng *et al.* present a post-processing method to enlarge the separation between wire segments to alleviate the speed-up effect [9, 10]. Meanwhile, to improve the routing area utilization, Tseng *et al.* shift long straight segments to reserve free space for critical wires requiring substantial length compensation [10]. However, increasing the segment separation necessitates a larger routing area, resulting in higher manufacturing costs.

Wu and Chao propose a different routing scheme depicted in Figure 1 (b), referred to as the *spiral delay line* [12]. In this alternate design, crosstalk accumulates asynchronously so that the noise can spread uniformly in time. Hence, the magnitude of the accumulated crosstalk is low, and the noise can be isolated from the main signal.

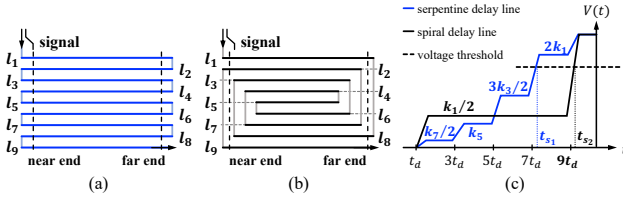


Figure 1: Illustration of nine-section (a) serpentine and (b) spiral delay lines. (c) Receiving waveforms of the nine-section serpentine and spiral delay lines: $V(t)$ denotes the accumulated crosstalk magnitude at the receiving end. The dashed line depicted a possible voltage threshold at the receiving end. t_{s1} and t_{s2} represent the time points at which the threshold is exceeded while utilizing serpentine and spiral delay lines, respectively.

Consequently, the transmitted waveform is of better integrity, and the crosstalk penalty is alleviated [7, 12].

The work proposed here aims to adopt spiral delay lines for length compensation to solve the delay matching problem. To this end, we mathematically model the geometric features of spiral delay lines to route wires of specific lengths that satisfy the delay-matching constraints. We also introduce a novel routing method that generates one wire at a time from the edge of the routing area toward its center. To optimize the allocation of routing resources, we develop a method to prioritize the free routing areas and select the areas with higher priorities to place the spiral delay lines. We implement our method by constructing a mixed-integer-linear programming (MILP) model to determine the wire's main path and a quadratic programming (QP) model to synthesize spirals for wire length compensation.

2 CROSSTALK NOISE

Every electrical signal has an electromagnetic field. *Crosstalk* is the electromagnetic interference (EMI) induced by one signal creating an undesired effect on another signal through overlapping electromagnetic fields. Due to the unintentional electromagnetic coupling between parallel wire segments, crosstalk is a common problem in PCB design.

In the following, we elaborate on the propagation of the main signal and the crosstalk along the serpentine and spiral delay lines. Figure 1 (a) and (b) illustrate examples of nine-section serpentine and spiral delay lines, respectively, where l_i denotes the i^{th} horizontal wire segment. These two designs differ in the way that the horizontal segments are connected. Specifically, in the serpentine delay lines in Figure 1 (a), the horizontal segments are sequentially connected at the near or far end, iteratively, e.g., l_1 is connected to l_2 at the far end, and then l_2 is connected to l_3 at the near end, and so on. As for the spiral design in Figure 1 (b), at the near end, l_2 is connected to the last segment l_9 , l_3 is connected to l_8 , and so on; at the far end, l_1 is connected to l_8 , l_2 is connected to l_7 , and so on.

Assume that the main signal is a one-volt ramped pulse and that the separation between two adjacent wire segments is identical. Since the coupling between two parallel wire segments depends primarily on their separation, the crosstalk pulse induced within the m^{th} and

n^{th} wire segments is of magnitude $\frac{k_{|m-n|}}{4}$, where $k_{|m-n|}$ denotes the average of the *capacitive* and *inductive coupling coefficients* separated by $|m-n|$ wire segments [7, 11]. It is well known that the crosstalk noise is inversely proportional to the wire segment separation; therefore, $k_{p_1} < k_{p_2}$ for $p_1 > p_2$. As a result, the most significant crosstalk noise is the crosstalk between two segments in the closest neighborhood, called the *adjacent crosstalk*. Moreover, all of the crosstalk pulses sustain a period of $2t_d$, where t_d indicates the time for a signal to travel along one wire segment [7, 11, 12].

2.1 The serpentine delay line

Suppose that at time zero, the main signal appears at the near end of l_1 . Due to the electromagnetic coupling, the signal simultaneously induces crosstalk noise in all other wire segments. Omitting the noise propagating toward the source, we focus on the leftward propagating crosstalk at the near ends of l_2 , l_4 , l_6 , and l_8 , which are pulses with a $2t_d$ width and $\frac{k_{|2-1|}}{4} = \frac{k_1}{4}$, $\frac{k_{|4-1|}}{4} = \frac{k_3}{4}$, $\frac{k_{|6-1|}}{4} = \frac{k_5}{4}$, and $\frac{k_{|8-1|}}{4} = \frac{k_7}{4}$ voltages, respectively [11]. As the signal propagates along l_1 , the crosstalk pulses propagate from l_2 , l_4 , l_6 , and l_8 to l_3 , l_5 , l_7 , and l_9 , respectively. At $t = t_d$, the main signal and crosstalk pulses reach the respective far ends. After that, the main signal travels along l_2 and induces additional rightward propagating crosstalk of magnitudes $\frac{k_{|3-2|}}{4} = \frac{k_1}{4}$, $\frac{k_{|5-2|}}{4} = \frac{k_3}{4}$, $\frac{k_{|7-2|}}{4} = \frac{k_5}{4}$, and $\frac{k_{|9-2|}}{4} = \frac{k_7}{4}$ at the far ends of l_3 , l_5 , l_7 , and l_9 , respectively. As a result, the receiver at the far end of l_9 receives a superposed crosstalk pulse spanning from t_d to $3t_d$ with a magnitude of $\frac{k_7}{4} + \frac{k_7}{4} = \frac{k_7}{2}$, which is shown by the first blue trapezoidal pulse in Figure 1 (c).

The process of crosstalk accumulation is repeated until the main signal reaches the receiver at $t = 9t_d$. The corresponding receiving waveform during this process is illustrated in Figure 1 (c). The main signal induces eight adjacent crosstalk pulses of magnitude $\frac{k_1}{4}$ before its arrival. These pulses simultaneously arrive at the receiving end at $t = 7t_d$ and are superposed as a large wave of magnitude $8 \times \frac{k_1}{4} = 2k_1$. This accumulation may exceed the threshold of logic switching at the receiving end before the main signal's arrival, as depicted in Figure 1 (c), leading to a speed-up effect on the wire [9, 10]. The influence of the crosstalk on the receiving waveform increases with the number of wire segments in the serpentine delay line [7, 11, 12].

2.2 The spiral delay line

In the spiral design shown in Figure 1 (b), the adjacent crosstalk pulse of magnitude $\frac{k_1}{4}$ induced by the main signal at l_1 at time zero proceeds straight to the receiver and appears in the time interval $[t_d, 3t_d]$. After time zero, the main signal propagates to the far end of l_8 and induces two adjacent crosstalk pulses of magnitude $\frac{k_1}{4}$ at $t = t_d$. The one at l_9 directly appears at the receiver, and the other one at l_7 requires an additional $2t_d$ to reach the receiver, spanning from $3t_d$ to $5t_d$. Similarly, at $t = 2t_d$, the main signal reaches the near end of l_3 and induces the adjacent crosstalk of magnitude $\frac{k_1}{4}$ at the near ends of l_2 and l_4 , spanning from $3t_d$ to $5t_d$ and $5t_d$ to $7t_d$, respectively. Consequently, in the time intervals $[t_d, 3t_d]$ and $[3t_d, 5t_d]$, the receiver presents a superposed crosstalk pulse of magnitude $\frac{k_1}{4} + \frac{k_1}{4} = \frac{k_1}{2}$.

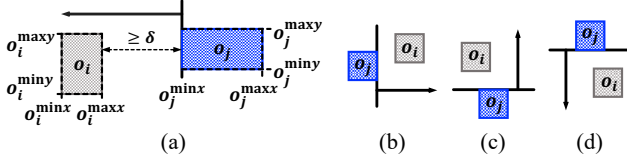


Figure 2: Illustration of the relative locations: o_i can be placed on the (a) left, (b) right, (c) top, or (d) bottom relative to o_j .

As with serpentine delay lines, the crosstalk noise is continuously created as the main signal propagates further toward the receiver. Still, these pulses reach the receiver at various time intervals, i.e., the crosstalk is accumulated asynchronously and spread over time ahead of the main signal. In particular, the magnitude of the crosstalk noise in the receiving waveform constantly stays at $\frac{k_1}{2}$, as illustrated in Figure 1 (c). Consequently, regardless of the number of delay line segments, the erroneous logic gate triggering does not occur as long as $\frac{k_1}{2}$ is smaller than the voltage threshold at the receiving end [12].

3 MATHEMATICAL MODEL

Our method generates one wire at a time. Each wire generation involves two phases: the *along-the-edge routing* and the length compensation utilizing spiral delay lines.

3.1 Modeling the area non-overlapping

Every electronic component is modeled in this work as a rectangular bounding box. In particular, wire segments are regarded as rectangles with their height or width set to zero. To avoid the overlapping of electronic components, we propose the following model.

Suppose that o_k is an arbitrary rectangle; we represent its position with the coordinates of its lower-left and upper-right corners, denoted as $(o_k^{\min x}, o_k^{\min y})$ and $(o_k^{\max x}, o_k^{\max y})$, respectively. Further, let $q_{i,j}^l, q_{i,j}^r, q_{i,j}^u,$ and $q_{i,j}^d$ be four auxiliary binary variables that indicate whether o_i is located on the left, right, top, or bottom relative to o_j , respectively. For instance, $q_{i,j}^l = 1$ represents that o_i is left to o_j , referring to the circumstance shown in Figure 2 (a), i.e., the distance between o_i 's right edge and o_j 's left edge should be greater than a minimally acceptable unit length δ . Thus, the x-coordinates of these two rectangles' edges should satisfy

$$o_i^{\max x} \leq o_j^{\min x} - \delta + (1 - q_{i,j}^l) \mathcal{M}, \quad (1)$$

where \mathcal{M} is an extremely large auxiliary constant. The linear constraints describing the relative location of o_i to the right, top, and bottom of o_j can be constructed analogously. Finally, if o_i does not overlap with o_j , o_i should be to the left, right, above, or below o_j . In other words, auxiliary binary variables $q_{i,j}^l, q_{i,j}^r, q_{i,j}^u,$ or $q_{i,j}^d$ must be set to 1.

$$q_{i,j}^l + q_{i,j}^r + q_{i,j}^u + q_{i,j}^d = 1. \quad (2)$$

3.2 Modeling the along-the-edge routing

The *center* of the routing area is typically densely populated with many wires, easily resulting in severe congestion. Considering this, we generate one wire at a time, starting from the edge of the

routing area and maximally reserving the center for unrouted wires. We introduce the MILP model to determine the wire's main path between its two target pins alongside the free space boundary in the following.

3.2.1 Modeling the bending point. A wire is formed by several vertical and horizontal segments. When two consecutive segments span in an orthogonal direction, a bend occurs, and the point of the bend is referred to as the *bending point*. To represent a wire, let its i^{th} segment, denoted by w_i , be the segment that starts at the i^{th} bending point, denoted as p_i , and ends at the $i + 1^{\text{th}}$ bending point p_{i+1} . We denote the length of w_i as l_i , and the coordinates of p_i as (x_i, y_i) . Further, we set one of the wire's target pins as the starting point and the other as the endpoint. Then, a wire can be considered as a path that starts at one pin and sequentially passes through each bending point to reach the other pin.

The possible spanning directions of w_i are represented by four binary variables, $s_i^l, s_i^r, s_i^u,$ and s_i^d , corresponding to the left, right, up, and down directions, respectively. The following equation ensures that each segment points to exactly one direction.

$$s_i^l + s_i^r + s_i^u + s_i^d = q_i, \quad (3)$$

where q_i is a binary variable that specifies whether the i^{th} bend exists or not. Particularly, two consecutive wire segments should not point to the opposite or the same direction, i.e.,

$$\begin{aligned} s_i^l + s_{i+1}^l + s_i^r + s_{i+1}^r &\leq 1, \\ s_i^u + s_{i+1}^u + s_i^d + s_{i+1}^d &\leq 1. \end{aligned} \quad (4)$$

Consequently, the coordinate of bending point p_{i+1} is expressed as

$$\begin{aligned} x_{i+1} &= x_i + l_i \cdot (s_i^r - s_i^l), \\ y_{i+1} &= y_i + l_i \cdot (s_i^u - s_i^d). \end{aligned} \quad (5)$$

We utilize the "big M method" [3] to linearize the product of variables and generate the following linear constraints to interpret $l_i \cdot s_i^l$ as an example. By considering the product $l_i \cdot s_i^l$ as a new integer variable, it can be described as

$$l_i \cdot s_i^l \leq l_i + (1 - s_i^l) \mathcal{M}, \quad l_i \cdot s_i^l \geq l_i - (1 - s_i^l) \mathcal{M}, \quad (6a)$$

$$l_i \cdot s_i^l \leq s_i^l \cdot \mathcal{M}. \quad (6b)$$

Specifically, if $s_i^l = 1$, the constraints in (6a) require $l_i \cdot s_i^l$ to be l_i ; otherwise, (6b) limits it to 0. The linear constraints characterizing $l_i \cdot s_i^r, l_i \cdot s_i^u,$ and $l_i \cdot s_i^d$ can be formulated analogously.

Further, to prevent the wire segment from intersecting other electronic components, we modify equation (1) as the following constraint to ensure that if $q_{i,j}^l$ is equal to 1, w_i is located on the left of component o_j .

$$x_i, x_{i+1} \leq o_j^{\min x} - \delta + (1 - q_{i,j}^l) \mathcal{M}. \quad (7)$$

The constraints about the right, top, and bottom relations between w_i and o_j can be derived analogously. Also, constraint (2) must be satisfied to ensure that the relative location between w_i and o_j matches one of the four situations shown in Figure 2.

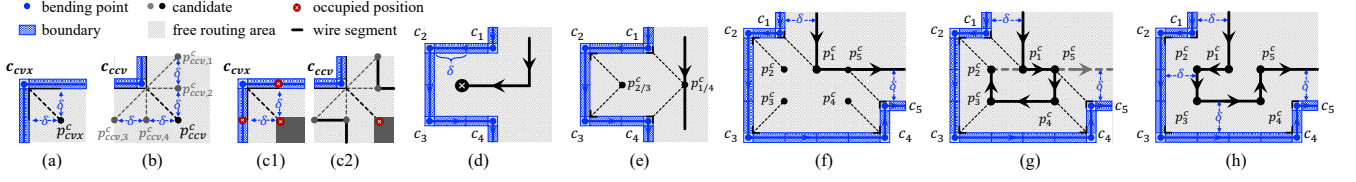


Figure 3: Illustration of the bending point candidates for (a) convex and (b) concave corners of the boundary. (c) Showcase occupying of the candidates by obstacles. (d) A misguided attempt in the narrow gap. (e) An expected path that bypasses the dead end. (f) A feasible but wasteful routing. (g) Showcase ignoring the order of bending point candidates, eventually leading to the infeasible routing sequence: $p_1^c - p_5^c - p_4^c - p_3^c - p_2^c$. (h) The along-the-edge routing.

3.2.2 Creating the bending point candidate. The along-the-edge routing determines the currently considered wire’s main path alongside the free space boundary. Specifically, the main path is delineated inside the free routing area along the area’s boundary, as illustrated in Figure 3 (h). Meanwhile, the most recently routed wire is the new boundary for the next wire generation.

Since vertical and horizontal segments form the wire, the free space boundary is a rectilinear polygon. A corner of the rectilinear polygon is *convex* or *concave* if the interior angle made by the two edges intersecting at this corner is 90° or 270° , respectively. When the wire is routed alongside the rectilinear polygon from the interior, only the corners cause the wire to bend. In other words, the point at a distance of the minimally acceptable unit length δ from the x - and y -coordinates of the polygon’s corner is most likely to be a bending point, referred to as p_{cvx}^c for the convex corner and p_{ccv}^c for the concave corner, as shown in Figure 3 (a) and (b), respectively. We define these points as the *bending point candidates*.

Moreover, if an obstacle is at the position of convex bending point candidate p_{cvx}^c , all points at a distance of one unit length from the x - or y -coordinates of corner c_{cvx} are occupied, as shown in Figure 3 (c1). Thus, we remove candidate p_{cvx}^c and do not create new candidates for that corner. As for a concave corner, suppose there is an obstacle at the position of point p_{ccv}^c , as shown in Figure 3 (c2). In that case, we create four alternative candidates, $p_{ccv,1}^c - p_{ccv,4}^c$, as illustrated in Figure 3 (b), to guarantee that the wire will bypass the obstacles.

3.2.3 Modeling the routing algorithm. Generally, our algorithm selects candidates to serve as bending points to form the wire’s main path. On one hand, if the wire bends at an inappropriate candidate, such as $p_{1/4}^c$ in Figure 3 (e), the routing path will be stuck in the dead end, as shown in Figure 3 (d). On the other hand, if not enough candidates are selected as bending points, as shown in Figure 3 (f), the wire will not be routed alongside the boundary, resulting in wasteful routing. Meanwhile, if the candidates are connected in the incorrect sequence, the circular wiring shown in Figure 3 (g) will occur. In the following, we introduce the modeling of our routing algorithm to generate a wire by sequentially connecting selected bending point candidates.

Let p_j^c denote the j^{th} candidate. We define a new binary variable that specifies whether the i^{th} bend is formed at p_j^c or not. Then, the following constraints establish the one-to-one correspondence

between candidates and bending points.

$$\sum_{1 \leq i \leq n_c} q_{i,j} \leq 1, \quad (8a) \quad \sum_{1 \leq j \leq n_c} q_{i,j} = q_i, \quad (8b)$$

where n_c denotes the number of candidates. The constraint in (8a) guarantees that the wire does not repeatedly pass through the same candidate, and the constraint in (8b) ensures that if a bend does occur, it should occur at a candidate.

Further, to prevent forming circular wires, the routing path of the currently considered wire should visit its bending points in the same order that the free space boundary (i.e., the most recently routed wire) visits its corners. To this end, we assign each bending point and each candidate a *connecting order*. We represent the connecting order of the i^{th} bending point with an integer variable r_i and the j^{th} candidate’s connecting order as a constant r_j^c equal to the index of the corresponding corner. Then, the following constraints should be satisfied if the i^{th} bend occurs at the j^{th} candidate.

$$x_i \leq x_j^c + (1 - q_{i,j}) M, \quad x_i \geq x_j^c - (1 - q_{i,j}) M, \quad (9a)$$

$$y_i \leq y_j^c + (1 - q_{i,j}) M, \quad y_i \geq y_j^c - (1 - q_{i,j}) M, \quad (9b)$$

$$r_i \leq r_j^c + (1 - q_{i,j}) M, \quad r_i \geq r_j^c - (1 - q_{i,j}) M, \quad (9c)$$

where (x_j^c, y_j^c) denotes the coordinate of p_j^c . Specifically, if $q_{i,j} = 1$, the constraints in (9a), (9b), and (9c) require the x - and y -coordinates, as well as the connecting orders, of p_i and p_j^c to be identical. Meanwhile, the following constraint is introduced to ensure the bending points are visited in the desired order.

$$r_i \leq r_{i+1}. \quad (10)$$

In order to make sure that there are enough bending points for the wire to be routed along the boundary, we maximize the number of selected bending points. Thus, the final optimization model is

$$\text{maximize: } \sum_{1 \leq i \leq n_c} q_i$$

Subject to: (3) – (10).

3.3 Modeling the length compensation

After determining the currently considered wire’s main path, we add spiral delay lines to address the delay-matching issue. In the following, we introduce the QP model for the spiral synthesis to compensate for the required length for the currently considered wire.

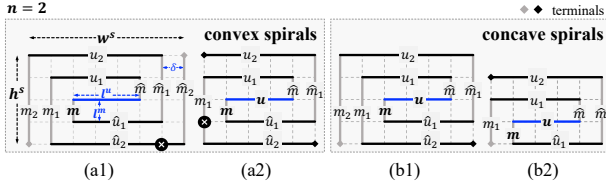


Figure 4: Illustration of (a) convex and (b) concave spirals.

3.3.1 *Modeling the spiral pattern.* As the orientation of the spiral's terminals can be either at opposite corners, as illustrated in Figure 4 (a), or on the same edge, as depicted in Figure 4 (b), we define two primary spiral types: *convex* and *concave* spirals.

As shown in Figure 4, each segment of the spirals, except for the blue segment, has another segment of the same length. Therefore, we call the blue segment the *unique segment*, indicated by u . The segment connected to the left end of u is referred to as the *minor segment*, represented by m . Meanwhile, u_i and m_i denote, respectively, the i^{th} parallel segments to u and m , from the pattern's center to its top and left. The segments of equal length to u_i and m_i are labeled as \hat{u}_i and \hat{m}_i , respectively.

Different subtypes of spirals are distinguished by the number of segments parallel to u and m . For convenience, let C^u and C^m be the sets of segments parallel to u and m , including u and m , respectively. As depicted in Figure 4 (a), the *cardinality* of C^u , denoted by $|C^u|$, for both subtypes of the convex spirals should be odd, indicated by $2n+1$, where n is an integer variable. Then, $|C^m|$ is expressed as $2(n+1)$ and $2n$ for the convex spirals in Figure 4 (a1) and (a2), respectively. In contrast, $|C^u|$ and $|C^m|$ are identical for both subtypes of the concave spirals, i.e., equal to $2n$ or $2n+1$. Finally, an arbitrary spiral can be constructed by rotating and flipping a convex or concave spiral subtype.

Further, let l^u and l^m be integer variables, representing the lengths of segments u and m , respectively. Then, the lengths of segments in C^u and C^m can be described as

$$l^{u_i} = l^u + (2i-1)\delta, \quad l^{m_i} = 2l^m + (2i-1)\delta, \quad (11)$$

respectively. As an example of the wire length calculation, the wire length of the convex spiral in Figure 4 (a1) can be stated as

$$L = l^u + 2 \sum_{1 \leq i \leq n} l^{u_i} + 2l^m + 2 \sum_{1 \leq i \leq n} l^{m_i} - 2\delta. \quad (12)$$

The concave spiral in Figure 4 (b1) can be considered as the convex spiral in Figure 4 (a1) with one outermost wire segment removed. Thus, its wire length can be expressed as

$$L = l^u + 2 \sum_{1 \leq i \leq n} l^{u_i} + 2 \sum_{1 \leq i \leq n} l^{m_i} - (2n+1)\delta. \quad (13)$$

The wire length of the other spiral types can be calculated analogously. Besides, the height h and width w of the convex spiral in Figure 4 (a1) can be described as

$$h = 2l^m + 2(n-1)\delta, \quad w = l^u + 2n\delta. \quad (14)$$

As for other spiral types, the expressions of height and width can be developed analogously.

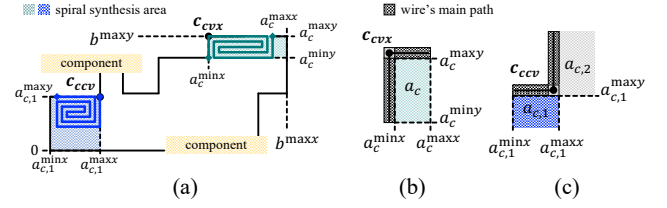


Figure 5: (a) An example of the spiral synthesis. Illustration of the spiral synthesis at (b) convex and (c) concave corners of wire's main path.

3.3.2 *Synthesizing spirals with prioritization.* Taking advantage of the rectangular form of the spiral's outer frame, we propose to use the spirals to fill in the nooks and crannies of the free routing area. To this end, we require that at least one of the spiral's corners overlap one of the boundary's corners, and the spiral's width and height should not exceed the length of the wire segments connected with that corner. Concretely, for the convex corner depicted in Figure 5 (b), a spiral can only be synthesized inside area a_c , and its top-left corner should overlap corner c_{cvx} with the coordinates $(a_c^{\min x}, a_c^{\max y})$. We define area a_c as the *spiral synthesis area* of convex corner c_{cvx} . We introduce the following constraints to synthesize a spiral at the convex corner c_{cvx} .

$$\begin{aligned} s^{\min x} &= a_c^{\min x}, & s^{\max x} &\leq a_c^{\max x}, \\ s^{\min y} &\geq a_c^{\min y}, & s^{\max y} &= a_c^{\max y}, \end{aligned} \quad (15)$$

where $s^{\min x}$, $s^{\max x}$, $s^{\min y}$, $s^{\max y}$, $a_c^{\min x}$, $a_c^{\max x}$, $a_c^{\min y}$, and $a_c^{\max y}$ are the x - and y -coordinate boundaries of spiral s and area a_c . As for the concave corner c_{ccv} portrayed in Figure 5 (c), areas $a_{c,1}$ and $a_{c,2}$ are available for synthesizing concave spirals. Without loss of generality, we introduce the synthesis of a spiral inside the spiral synthesis area $a_{c,1}$ of the concave corner c_{ccv} . Contrary to the convex case, the two edges intersecting at the corner c_{ccv} cannot specify the height of area $a_{c,1}$. Thus, to avoid creating a towering spiral that easily splits the free space, the spiral's height h is set to be smaller than its width w .

$$\begin{aligned} s^{\min x} &\geq a_{c,1}^{\min x}, & s^{\max x} &= a_{c,1}^{\max x}, \\ s^{\max y} &= a_{c,1}^{\max y}, & h &\leq w. \end{aligned} \quad (16)$$

Combining constraints (15) and (16) with constraint (14), the values of integer variables l^u , l^m , and n can be determined. After that, using the wire length calculation presented in Section 3.3.1, we obtain the wire lengths of the synthesized spirals. Further, since the compensated length of a spiral, denoted by ΔL , is the length difference between the spiral and the wire's main path, ΔL is obtained by subtracting the spiral's height or width from L . Specifically, the compensated length of the convex spiral synthesized at corner c_{cvx} in Figure 5 (b) is $L - (h + w)$, while that of the concave spiral synthesized at corner c_{ccv} in Figure 5 (c) is $L - w$.

Let ΔL_w denote the required compensated length of the currently considered wire. Then, we introduce the following in-equation for

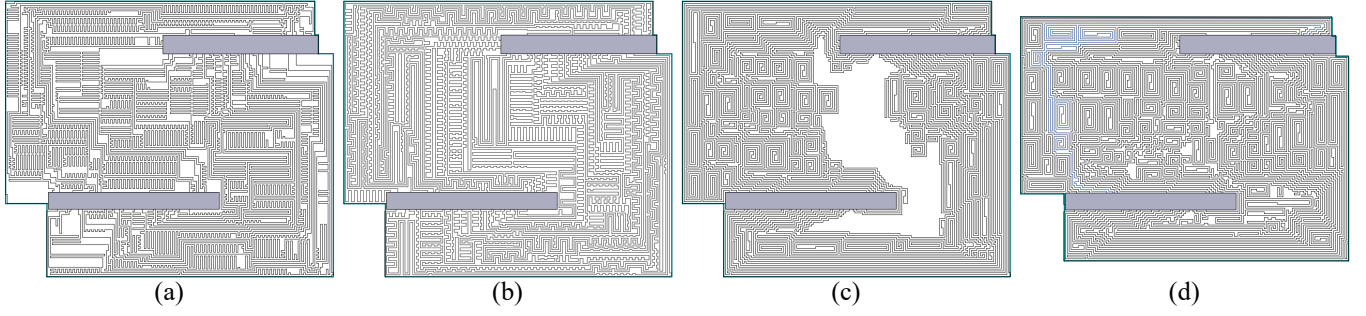


Figure 6: Routing results of case 6. (a) Original result in [14]. (b) Routing result using the method in [10]. Our routing results in (c) the original routing area, and (d) the minimized routing area.

the length matching.

$$\sum_{1 \leq k \leq n_c} \Delta L_k \cdot q_k^s \leq \Delta L_w, \quad (17)$$

where q_k^s is a binary variable that specifies whether a spiral is synthesized at the k^{th} corner, and ΔL_k denotes the corresponding compensated length. In the above constraint, using inequality instead of equality avoids infeasible solutions during the solving process. The equality is achieved by setting the objective function to maximize the compensated wire length.

To prevent overlap between two spirals, we modify in-equation (1) into the following constraint, indicating that s_i is located on the left of s_j .

$$s_i^{\max x} \leq s_j^{\min x} - \delta + (1 - q_{i,j}^l)M. \quad (18)$$

Analogously, the constraints of the right, top, and bottom relations between s_i and s_j can be constructed. Also, constraint (2) is required to ensure the non-overlapping between spirals.

Further, to improve the allocation of free routing areas, we provide a weight coefficient in the range of $[0, 1]$ for each boundary's corner to indicate a priority for synthesis. Each spiral synthesis area a with $a^{\min x}$, $a^{\max x}$, $a^{\min y}$, and $a^{\max y}$ as x - and y -coordinate boundaries can be expressed as the intersection of two relevant rectangles on the circuit board, i.e.,

$$a = [a^{\min x}, a^{\max x}] \times [0, b^{\max y}] \cap [0, b^{\max x}] \times [a^{\min y}, a^{\max y}],$$

where $b^{\max x}$ and $b^{\max y}$, as shown in Figure 5 (a), denote the upper boundaries of the circuit board's x - and y -coordinates, respectively. Only the free space within the above-indicated relevant rectangles is occupied when the spiral is synthesized at the corresponding corner. Thus, to reflect the severity of routing congestion after the spiral synthesis in this corner, the corresponding weight coefficient is calculated as the area ratio of the free space within the relevant rectangles to the union of these two relevant rectangles.

As mentioned in Section 2.2, the magnitude of the crosstalk induced by an arbitrary spiral stays constant regardless of the number of that spiral's wire segments. Thus, the accumulated crosstalk of one wire depends mainly on the number of spirals on that wire. Finally, to compensate for the required wire length while minimizing

the crosstalk penalty, the objective function is expressed as

$$\text{maximize: } \sum_{1 \leq k \leq n_c} (\alpha_k \Delta L_k - q_k^s),$$

$$\text{Subject to: } (11) - (18),$$

where α_k indicates the weight coefficient for the k^{th} corner.

4 EXPERIMENTAL RESULTS

In this section, we investigate the performance of the along-the-edge routing model and our length compensation technique using six layout designs as test cases. Cases 1–5 are from [1], while case 6 is an example of the BSG routing proposed in [14]. Our work is implemented using Java, and the experiments are performed using a computer with an Apple M1 8-core CPU. The MILP and QP models are solved by Gurobi [5].

4.1 Routing area reduction

The routing results for case 6 are illustrated in Figure 6, where the green rectilinear polygon is the boundary of the routing area. Figure 6 (a) shows the routing results proposed by Yan *et al.*, applying serpentine delay lines for delay matching [14]. Figure 6 (b) shows the routing results proposed by Tseng *et al.* with wider meander segments [10]. Specifically, the separation between parallel wire segments is 1.6 times larger. We can see that the results in [10] also have a more uniform wire distribution than in [14].

Our result, employing spiral delay lines for delay matching, is illustrated in Figure 6 (c). The figure demonstrates that the nooks and crannies of the routing area are filled by adding spirals, and a large surplus of free space remains in the central area. To quantify the reduction in routing area requirements using our method, we decrease the height and width of the original routing area and reroute within the minimized routing area. The final routing result is illustrated in Figure 6 (d). As shown in the figure, the height of the routing area is significantly reduced, and there is almost no wasted routing space. Compared to the original design in [14], we reduce the required routing area by 10.99%.

Table 1 demonstrates the routing efficiency of our method. In our experiments, the minimally acceptable unit length δ is set to be 1. Supposing the wire segment separation is one unit length,

Table 1: Routing efficiency comparison

case	n_w^1	l_{tot}^2	time ³ (s)	a_Δ^4 (%)	\hat{a}_Δ^5 (%)	R_a^6 (%)
1	12	448	5	71.82	14.99	66.86
2	16	2856	8	43.52	17.10	31.88
3	16	3046	10	42.40	14.12	32.92
4	16	5332	32	36.57	12.90	27.17
5	20	3802	26	32.50	16.77	18.91
6	36	19393	950	22.83	13.30	10.99

1. the number of wires, 2. the total required wire length, 3. the runtime, 4. and 5. the free space ratio of the original and minimized routing area, respectively, 6. the percentage routing area reduction.

then adding one unit-length wire segment is equivalent to occupying one square of area $\delta^2 = 1$. Therefore, to compensate for the required wire length, denoted by l_{tot} , the routing area should be at least l_{tot} . In other words, without considering the feasibility of the routing result, the maximum routing area reduction is the difference between the total routing area and l_{tot} . The percentage ratio of this difference in the total routing area, defined as the *free space ratio*, evaluates the utilization of the routing area. Specifically, a smaller free space ratio indicates less wasted area. As demonstrated in Table 1, our method can effectively reduce the free space ratio to less than 18% in all cases. On average, the required routing area decreases by 31% compared to the original routing result, leading to lower manufacturing costs.

Further, as shown in the table, all test cases are solved in less than 16 minutes. In particular, for cases 1–5, each case was solved within a minute. We conclude that the proposed method has high efficiency. Generally, for the same number of wires, the longer the required total wire length, the longer the time needed. Our method spent the longest time obtaining the routing result for case 6. This occurs because we limit the routing area to a small value so that wires have to bend frequently to meet the length compensation requirements, leading to more variables in the along-the-edge routing model and, thus, longer time for finding the optimal solution.

4.2 Quantitative analysis of the crosstalk noise

4.2.1 Crosstalk evaluation regarding different segment separation.

To evaluate the accumulated crosstalk using different routing methods, we demonstrate the following model to describe the adjacent crosstalk with various segment separations quantitatively. Consider the most elementary instance of the two-section serpentine line, equivalent to the two parallel wire segments depicted in Figure 7 (a). As mentioned in Section 2, t_d is the propagation delay of one wire segment. Further, let $V_A(t)$ be a ramped pulse at the sending end of the active line, whose rise time t_r is assumed to be smaller than the round-trip time $2t_d$, as shown in Figure 7 (b). Then, the general expression for the crosstalk at the near end is [2, 7, 11]

$$V_{NE}(t) = K_{NE} \cdot (V_A(t) - V_A(t - 2t_d)), \quad (19)$$

where the coupling coefficient K_{NE} is [7, 11]

$$K_{NE} = \frac{1}{4} \left(\frac{|C_m|}{C_s} + \frac{L_m}{L_s} \right), \quad (20)$$

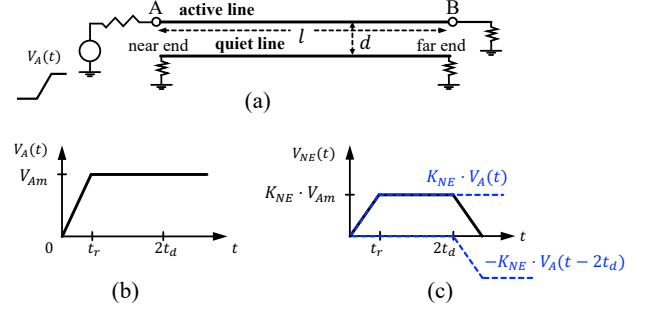


Figure 7: Illustration of the crosstalk noise of two coupled wire segments: (a) Two parallel segments with matched terminations. (b) The voltage function at the sending end of the active line. (c) The waveform of the crosstalk at the near end.

and C_m and L_m are the *mutual capacitance* and *mutual inductance*, respectively, and C_s and L_s are the *self capacitance* and *self inductance*, respectively. Based on equation (19), we illustrate the corresponding waveform at the near end in Figure 7 (c). The near-end crosstalk is observed to reach the saturated value of $K_{NE} \cdot V_{Am}$, where V_{Am} denotes the voltage of the main signal. In other words, the value of the coupling coefficient K_{NE} , which mainly depends on the separation of the wire segments, determines the magnitude of the crosstalk for a certain V_{Am} .

To estimate the coupling coefficient K_{NE} , we introduce the following expressions for the capacitance and inductance of two parallel wire segments [4, 8].

$$C_m = \frac{\pi \epsilon l}{\cosh^{-1} \left(\frac{d}{2r} \right)}, \quad C_s \propto l, \quad (21)$$

$$L_m = 2l \left(\ln \left(\frac{2l}{d} \right) - 1 + \frac{d}{l} \right), \quad L_s = \frac{\mu_0}{\pi} l \cdot \cosh^{-1} \left(\frac{d}{2r} \right),$$

where ϵ is the *absolute permittivity*, l is the wire segment length, r is the wire radius, d is the wire segment separation, and $\mu_0 \approx 4\pi \times 10^{-7}$ H/m is the *magnetic permeability of free space*. Since l is not significantly greater than d in our case, the coupling coefficient for the near-end crosstalk induced within two parallel wire segments separated by d is estimated as

$$k_d \approx \frac{\cosh^{-1} \left(\frac{d}{2r} \right)}{\cosh^{-1} \left(\frac{\delta}{2r} \right)} \cdot k_\delta, \quad (22)$$

where δ is the one unit length and k_δ represents the average of the capacitive and inductive coupling coefficients between two parallel segments with one unit-length separation. To compare the accumulated crosstalk of the routing results with different segment separations, we utilize equation (22) to express the magnitude of the accumulated adjacent crosstalk for an arbitrary delay pattern in k_δ as a unit.

In our experiments, the wire segment separations in the routing results for case 6 adopting methods [14], [10], and the proposed method in the original and minimized routing areas are 1, 1.6, 1, and 1, respectively. Since the wire radius is significantly smaller than

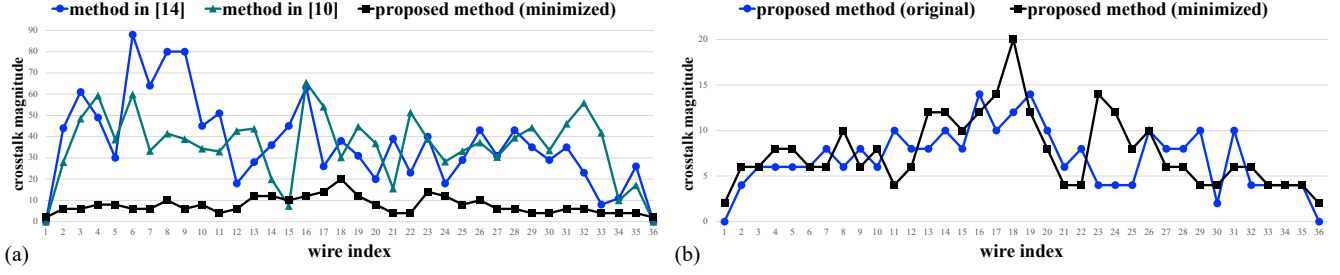


Figure 8: Illustration of the magnitude of the accumulated crosstalk noise using different methods for each wire in case 6.

Table 2: Crosstalk magnitude comparison of case 6

		V_{max}	V_{avg}
method in	[14]	88.00	36.94
	[10]	65.43	35.66
proposed method in	original area	14.00	6.94
	minimized area	20.00	7.61

the segment separation, r is set to be 0.0001. The above equations imply that $k_{1,6}$ equals $0.95 \cdot k_1$.

Further, let the main signal be a ramped pulse of unit voltage. Based on the elaboration in Section 2, we conclude that the superposed adjacent crosstalk of an n -section serpentine and spiral delay lines with the separation d are of voltages

$$V_{serpentine} = (n-1) \cdot \frac{kd}{4}, \quad V_{spiral} = \frac{kd}{2}, \quad (23)$$

respectively. Noticeably, when adopting serpentine delay lines, the accumulated crosstalk increases as the number of delay line segments increases. In contrast, the magnitude of the crosstalk remains unchanged for the spiral design with a different number of wire segments.

As explained in Section 2, the width of the superposed crosstalk pulse is $2t_d$, where t_d equals the product of the delay pattern's wire segment length and the per-unit-length propagation delay. Therefore, the induced crosstalk noise of the delay patterns with extremely short segment lengths does not accumulate together with the crosstalk of other patterns. Considering this, we classified the delay patterns into two types: the delay patterns with a wire segment length $> 2\delta$ and $\leq 2\delta$. After that, the crosstalk magnitude of each wire is determined as

$$\max \{xtalk_{>2\delta}, xtalk_{\leq 2\delta}\}, \quad (24)$$

where $xtalk_{>2\delta}$ and $xtalk_{\leq 2\delta}$ describe the accumulated crosstalk magnitudes of the delay patterns with segment lengths $> 2\delta$ and $\leq 2\delta$, respectively.

4.2.2 Crosstalk magnitude comparison of case 6. According to the crosstalk evaluation described in 4.2.1, we calculated the crosstalk magnitude of each wire for each routing result. The calculation results are presented in Table 2, where V_{max} and V_{avg} denote the maximum and average crosstalk magnitudes among 36 wires, respectively. The values in the table are expressed in units of $\frac{k_1}{4}$.

Observing the magnitudes of the accumulated crosstalk for the routing results in [10] and [14] indicates that, by enlarging the separation between parallel wire segments, the accumulated crosstalk is effectively alleviated using the method in [10]. Specifically, the maximum and average magnitudes of the crosstalk noise are reduced by 25.65% and 3.47%, respectively.

As for the proposed method, while routing in the original routing area, referring to the routing result in Figure 6 (c), our method significantly decreases both the maximum and average magnitudes of the crosstalk noise. Compared to the methods in [14] and [10], the maximum crosstalk magnitude of our routing result is decreased by 84.09% and 78.60%, respectively, while the average magnitude is decreased by 81.21% and 80.54%, respectively.

When the total routing areas are reduced, the shrunk space increases the number of spirals and, consequently, the crosstalk magnitude. Still, compared to the original design in [14], our routing result's maximum and average crosstalk levels decreased by 77.27% and 79.40%, respectively. Meanwhile, compared to the result in [10], the crosstalk penalty decreases by at least 69%. In other words, our method can still significantly reduce the disturbance caused by crosstalk noise in designs with a small free space ratio.

Further, we present the crosstalk magnitude accumulated within each wire for case 6 in Figure 8 (a). The figure demonstrates that the crosstalk noise in our routing results is significantly lower than that accumulated by other methods for almost every wire. Besides, we illustrated the crosstalk magnitude of each wire in our results routing in the original and minimized routing areas in Figure 8 (b). The line chart demonstrates that the crosstalk noise level is not notably impacted while the routing area is decreased. In conclusion, our method is reliable in alleviating the accumulated crosstalk noise.

5 CONCLUSIONS

In this work, we proposed a routing method that mitigates the speed-up effect while decreasing the routing area requirements by utilizing spiral delay lines for delay matching. Our routing algorithm was proposed to generate one wire at a time along the edge of the routing area. Meanwhile, we proposed a method to synthesize the spirals for each wire to compensate for the required wire length. The wire generation and the spiral synthesis were implemented by constructing an MILP and a QP model, respectively. Experimental results confirm that the proposed method can significantly decrease the required routing area and the accumulated crosstalk noise.

REFERENCES

- [1] Ching-Yu Chin, Chung-Yi Kuan, Tsung-Ying Tsai, Hung-Ming Chen, and Yoji Kajitani. 2013. Escaped Boundary Pins Routing for High-Speed Boards. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32, 3 (2013), 381–391. <https://doi.org/10.1109/TCAD.2012.2221714>
- [2] A. Feller, H. R. Kaupp, and J. J. Digiacomio. 1965. Crosstalk and Reflections in High-Speed Digital Systems. In *Proceedings of the November 30–December 1, 1965, Fall Joint Computer Conference, Part I (Las Vegas, Nevada) (AFIPS '65 (Fall, part I))*. Association for Computing Machinery, New York, NY, USA, 511–525. <https://doi.org/10.1145/1463891.1463947>
- [3] Igor Griva, Stephen G Nash, and Ariela Sofer. 2009. *Linear and Nonlinear Optimization: Second Edition*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).
- [4] John David Jackson. 1998. *Classical Electrodynamics*. Wiley New York.
- [5] Gurobi Optimization LLC. 2022. Gurobi Optimizer Reference Manual. <https://www.gurobi.com>
- [6] Muhammet Mustafa Ozdal and Martin D. F. Wong. 2006. A Length-Matching Routing Algorithm for High-Performance Printed Circuit Boards. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25, 12 (2006), 2784–2794. <https://doi.org/10.1109/TCAD.2006.882584>
- [7] Omar Ramahi. 2003. Analysis of Conventional and Novel Delay Lines: A Numerical Study. *The Applied Computational Electromagnetics Society Journal (ACES)* 18 (11 2003).
- [8] Edward Bennett Rosa. 1908. *The self and mutual inductances of linear conductors*. Number 80 in Bulletin of the Bureau of Standards. US Department of Commerce and Labor, Bureau of Standards. 301–344 pages.
- [9] Tsun-Ming Tseng, Bing Li, Tsung-Yi Ho, and Ulf Schlichtmann. 2013. Post-route alleviation of dense meander segments in high-performance printed circuit boards. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 713–720. <https://doi.org/10.1109/ICCAD.2013.6691193>
- [10] Tsun-Ming Tseng, Bing Li, Tsung-Yi Ho, and Ulf Schlichtmann. 2015. ILP-Based Alleviation of Dense Meander Segments With Prioritized Shifting and Progressive Fixing in PCB Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 6 (2015), 1000–1013. <https://doi.org/10.1109/TCAD.2015.2402657>
- [11] Ruey-Beei Wu and Fang-Lin Chao. 1995. Laddering wave in serpentine delay line. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B* 18, 4 (1995), 644–650. <https://doi.org/10.1109/96.475270>
- [12] Ruey-Beei Wu and Fang-Lin Chao. 1996. Flat spiral delay line design with minimum crosstalk penalty. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B* 19, 2 (1996), 397–402. <https://doi.org/10.1109/96.496044>
- [13] Tan Yan and Martin D. F. Wong. 2008. BSG-Route: A length-matching router for general topology. In *2008 IEEE/ACM International Conference on Computer-Aided Design*. 499–505. <https://doi.org/10.1109/ICCAD.2008.4681621>
- [14] Tan Yan and Martin D. F. Wong. 2009. BSG-Route: A Length-Constrained Routing Scheme for General Planar Topology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28, 11 (2009), 1679–1690. <https://doi.org/10.1109/TCAD.2009.2030352>