An Efficient Two-Phase ILP-Based Algorithm for Precise CMOS RFIC Layout Generation

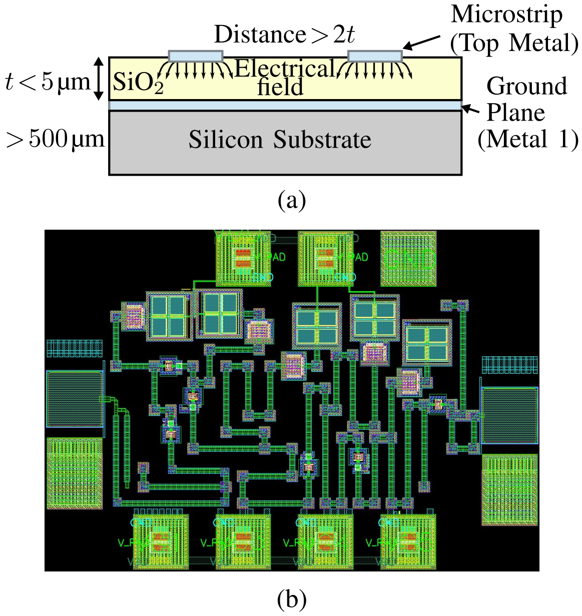
Tsun-Ming Tseng, *Member, IEEE*, Bing Li, Ching-Feng Yeh, Hsiang-Chieh Jhan,
Zuo-Min Tsai, *Member, IEEE*, Mark Po-Hung Lin, *Senior Member, IEEE*, and Ulf Schlichtmann, *Member, IEEE*

***Abstract*—With advancing process technologies and booming IoT markets, millimeter-wave CMOS RFICs have evolved rapidly and been widely applied in recent years. The performance of CMOS RFICs is very sensitive to the chip layout, and a tiny variation of the microstrip length can cause a large impact to the circuit performance. This results in a time-consuming tuning process including much simulation effort for chip design, which becomes the major bottleneck for time to market. This paper introduces a progressive integer-linear-programming-based method consisting of two phases: global layout generation and iterative validation. In the global layout generation phase, we focus on the most critical constraints such as layout planarity and device connection relations to determine the topology of the final design. This provides a basis for constructing the accurate model in the iterative validation phase. The layouts generated by applying our method can satisfy very stringent routing requirements of microstrip lines, including spacing/non-crossing rules, precise length, and bend number minimization, within a given layout area. The resulting RFIC layouts excel in both performance and area with much fewer bends compared with the simulation-tuning based manual layout, while the layout generation time is significantly reduced from weeks to a few minutes.**

I. Introduction

In wireless communication systems, RFICs are key components to receive or transmit RF signals. Millimeter-wave (mm-wave) RFICs based on CMOS process technologies have become more and more popular due to cost-effective and power-efficient system-on-chip integrations [1], [2]. Although RFICs only contain a few transistors and some passive components, such as capacitors, inductors, and transmission lines, a high-quality layout is essential as the circuit performance is very sensitive to the circuit layout, in contrast to many digital and analog designs.

In order to implement transmission lines based on CMOS technologies, thin-film microstrip lines [3], as demonstrated in Figure 1(a), are commonly adopted. A microstrip line and its ground plane are usually implemented with the top metal layer and the bottom metal layer (i.e. Metal 1), respectively. Due to the shielding of the ground plane, the lossy silicon substrate does not cause signal loss to transmission lines. As the distance, t, between the microstrip and its ground plane is small, which is about 5µm for 90nm CMOS technologies, the coupling effect between two microstrip lines can be neglected if the distance between them is larger than $2t$ [3], [4], or 10µm for 90nm CMOS technologies. In addition to the spacing rule, any crossing between microstrip lines is not allowed. The routing of all microstrip lines must be planar.



(a)

(b)

Figure 1: (a) The cross section of microstrip lines. (b) A manually designed CMOS RFIC layout (890µm×615µm) of a 94 GHz LNA with planar routing of all microstrip transmission lines.

To achieve good RF circuit performance, layout design of RFICs, especially the routing of all microstrip lines, is extremely critical. In addition to the aforementioned spacing and non-crossing rules, any increment/decrement in length or routing bends of a microstrip line may have negative impact on circuit performance [5]. Consequently, the layout design of mm-wave CMOS RFICs has been a labor-intensive and time-consuming task. Since it is very difficult for humans to generate an exact layout of an RFIC within a restricted

The preliminary version of this paper was published in the Proceedings of the 53rd Annual Design Automation Conference (DAC), 2016.

This work was partially supported by Alexander von Humboldt Foundation of Germany, and the Ministry of Science and Technology (MOST) of Taiwan, under Grant Nos. 104-2628-E-194-001-MY3, 105-2221-E-194-010, and 105-3011-E-002-002.

Tsun-Ming Tseng, Bing Li, and Ulf Schlichtmann are with the Institute for Electronic Design Automation, Technical University of Munich (TUM), Munich 80333, Germany (e-mail: tsun-ming.tseng@tum.de; b.li@tum.de; ulf.schlichtmann@tum.de).

Ching-Feng Yeh, Hsiang-Chieh Jhan, Zuo-Min Tsai, and Mark Po-Hung Lin are with the Department of Electrical Engineering and the Advanced Institute of Manufacturing with High-tech Innovations, National Chung Cheng University, Chiayi 621, Taiwan (e-mail: craig20050122@gmail.com; lu1833@gmail.com; zuomintsai@ccu.edu.tw; marklin@ccu.edu.tw).

TCAD, DOI: 10.1109/TCAD.2016.2629445
http://ieeexplore.ieee.org/document/7745860/

layout area, designers first generate a rough initial planar layout followed by very tedious iterative simulation tuning and circuit/layout refinement. Each iteration includes i) performing full-wave electromagnetic (EM) simulation; ii) resizing devices/microstrips according to the simulation results and designers' experience; iii) adjusting the respective layout. Such manual procedure requires a large number of iterations. Experienced designers would even spend more than two weeks to finish a satisfactory layout of a 94 GHz low-noise amplifier (LNA), as shown in Figure 1(b).

A. *Previous Work*

Only few studies [6], [7] in the literature proposed automatic RFIC layout generation methods. Actuna *et al.* [6] focused on floorplanning, while they suggested to perform gridless maze routing afterwards. Such separation between floorplanning and routing is not suitable for CMOS RFICs with microstrip lines which need to be planar and precise. Pathak and Lim [7] presented a methodology to automatically generate RFIC layouts by iteratively performing placement and routing, and resizing circuit components to compensate performance degradation due to imprecise routing.

On length matching of wires for conventional ICs and printed circuit boards (PCBs), [8]–[11] tried to minimize either length difference or length-ratio difference among a set of nets during routing. Such problem formulation cannot meet the stringent routing requirements of microstrip lines in mm-wave CMOS RFICs because the length of each microstrip line after routing must be exactly the same as the given length at the circuit design to maintain the expected RF circuit performance. Recently, [12], [13] are proposed to precisely match wire lengths to given values, but similar to [8]–[11], these routing methods all assume that devices are not movable, and hence fail to generate precise lengths of all microstrip lines.

B. *Our Contributions*

Different from the previous work [6], [7], which did not focus on length precision and bend number minimization of microstrip lines during RFIC layout generation, we propose a better layout generation methodology with an emphasis on microstrip routing optimization resulting in better performance matching before and after layout design. Our contributions are summarized below:

- We comprehensively introduce the essential routing considerations of microstrip lines, including spacing/non-crossing rules, precise length, bend smoothing, bend number minimization, and equivalent length modelling of bends;
- According to the routing considerations, we present a new problem formulation to generate the layout with precise placement and routing within a given layout area while exactly matching microstrip lengths to the given values and minimizing the number of bends;
- Based on the problem formulation, we establish a complete integer-linear-programing (ILP) model for concurrent exact placement and routing. The potential routing bends on microstrips are modeled by introducing chain points;
- In order to simplify the sophisticated ILP model, we further propose an improved progressive ILP-based (IP-ILP) RFIC layout generation method, which starts within a global layout generation phase estimating suitable locations for devices and to route microstrips. An interative validation phase is followed to generate a final layout satisfying all design rules.
- Compared with manual layout design, given the same or even smaller layout area, the proposed IP-ILP method can reduce RFIC layout design time from weeks to several minutes, and result in even better circuit performance with much fewer bends on microstrip lines.
- Compared with the progressive ILP-based (P-ILP) method proposed in our preliminary work [14], the layouts generated by applying the new method have better performance within drastically reduced program runtime.

The structure of this paper is organized as follows. Section II introduces microstrip routing considerations. Section III presents the problem formulation. Section IV describes a general ILP model for exact placement and routing of devices and microstrips. Section V proposes a novel progressive RFIC layout generation method based on the ILP model. Finally, experimental results and conclusions are given in Sections VI and VII, respectively.

II. Microstrip Routing Considerations

Before introducing our problem formulation, the most important routing considerations for microstrip lines should be clarified.

A. *Coupling Effect*

Spacing distance between devices such as NMOSs and I/O pads is always satisfied, since their dimensions in the circuit specification typically include the required spacing distance according to their corresponding device types. The dimensions of these devices can be considered as expanded bounding boxes containing real-size devices.

However, the spacing distance between microstrips cannot be directly satisfied in the same manner, since the shape of microstrips can be other than rectangles, which is difficult to be directly expanded as bounding boxes. In order to ensure the quality of signals, the minimum spacing between two microstrips should be $2t$ to reduce coupling [3], [4], where t is the distance between the microstrip and its ground plane, as mentioned in Section I. In our method, we first decompose a microstrip line into several segments with the microstrip bends as their ends. We call these ends *chain points* in our method and denote them by their coordinates, e.g. (x_1, y_1), (x_2, y_2), and (x_3, y_3) in Figure 2(a).

Then we create bounding boxes around each device and each segment, expanding their horizontal and vertical dimensions by t on both sides. For each pair of bounding boxes that does not contain consecutive segments, we keep them from overlapping as illustrated in Figure 2(b), where all spacings among devices and microstrip lines are ensured. The expanded bounding boxes are used to describe the overlapping constraints in our ILP model afterwards. This expansion can

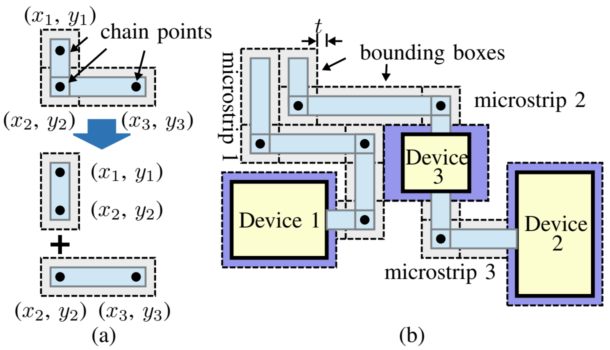


Figure 2: (a) Microstrip decomposition at chain points. (b) Expanded bounding boxes of devices and microstrips for satisfying spacing rules due to the coupling effect.

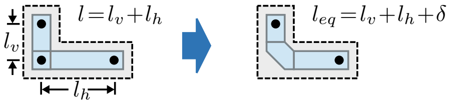


Figure 3: Bend smoothing due to discontinuity effects and equivalent length modelling.

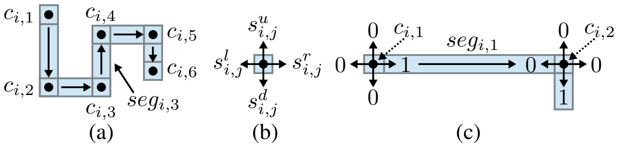


Figure 4: (a) An example of a microstrip model with 6 chain points and 5 segments. (b) Binary directional variables indicating spanning directions at chain points. (c) An example of a segment $seg_{i,1}$ spanning from left to right.

be easily extended to cover cases with different spacing rules among microstrips and devices.

B. Discontinuity Effect

Bends on a transmission line may cause signal loss, which is the major source of discontinuity effects [5]. To mitigate the discontinuity effect, it is essential to minimize the number of bends on microstrip lines. As the bends are sometimes unavoidable due to limited layout area, we model potential microstrip bends by chain points, e.g. the chain point at (x_2, y_2) as shown in Figure 2(a).

In addition to minimizing the number of bends on microstrip lines, any 90° bend must be smoothed, or replaced by a diagonal shortcut, as demonstrated in Figure 3, to reduce signal loss. Such transformation results in a different microstrip length for signal propagation, which cannot be directly represented by its geometrical length. Instead, an equivalent length change, δ, must be calculated by RF simulation of the diagonal bend and comparing the signal propagation with the case through a straight microstrip. In other words, each time when a signal goes through this diagonal bend, the propagation characteristics are equivalent to the case that it goes through a straight microstrip with the equivalent length, $l_{eq} = l_v + l_h + \delta$.

Since δ is typically a negative value, microstrips with more bends need to be routed longer to meet the designated length. However, as one of the major difficulties in length-matching problems is that the routing space is limited, the longer the routing lines are, the more difficult the problem will become. For 90nm CMOS technologies, δ is around 5µm and seems small. However, if we take the LNA design shown in Figure 1(b) as an example, the total number of bends is 59, and the length compensation owing to bends can be up to 5×59=295µm. This length is a burden for routing, considering the dimension of the chip. Therefore, the reduction of bends is meaningful. Note that other patterns similar to the diagonal bend in Figure 3 can also be used for transmission line smoothing. The method discussed in the following sections can be adapted easily to incorporate these patterns into the proposed method.

III. Problem Formulation

To generate a layout for an RFIC with precise placement and routing, the input, constraints and output are detailed below.

Input: i) The netlist of the circuit; ii) The dimensions of the layout area; iii) The dimensions of devices; iv) The width of microstrips; v) The required distance between microstrip segments and/or devices; vi) The equivalent length compensation, δ, for a smoothed bend; vii) The exact lengths of all microstrip lines.

Constraints: i) The equivalent lengths of microstrips should be equal to the given values; ii) No overlap exists among microstrip segments and/or devices due to the planar routing requirement; iii) Pads should be placed at the boundary of the layout area.

Output: A layout with a minimized number of microstrip bends.

IV. ILP Model for Exact Placement and Routing

We first describe a precise ILP model for concurrent exact placement and routing, which includes the microstrip model with the emphasis on minimizing the number of bends. We consider device flip and rotation, pin-location switching, and their impact on connections among devices and microstrips. The optimization objective of this precise model is proposed at the end of this section.

A. Microstrip Model

A microstrip line can be decomposed into several horizontal and vertical segments with the chain points as their ends as mentioned in Section II. An example of a microstrip line consisting of 6 chain points and 5 segments is shown in Figure 4(a).

In order to determine the location and the routing pattern of a microstrip line, we need to determine the location, the spanning direction, and the length of each segment that forms the corresponding microstrip line. In our method, the required information of each segment is represented by the chain point at the segment tail.

1) Spanning direction

Assume that there are m microstrips in total in the circuit and the i^{th} microstrip has n_i chain points, meaning that this microstrip is formed by n_i-1 consecutive segments. We denote the j^{th} chain point in the i^{th} microstrip as $c_{i,j}$, and the coordinate of $c_{i,j}$ as $(x_{i,j}, y_{i,j})$, where $j=1,\ldots n_i$. With above specification, we represent the possible directions of the segment $seg_{i,j}$ that starts from $c_{i,j}$ by four binary directional variables $s^l_{i,j}$, $s^d_{i,j}$, $s^r_{i,j}$, and $s^u_{i,j}$, corresponding to the left, down, right, and up directions in a two-dimensional space, as illustrated in Figure 4(b). In the example shown in Figure 4(c), directional variables $s^u_{i,1}$, $s^d_{i,1}$, and $s^l_{i,1}$ are set as 0, and $s^r_{i,1}$ is set to 1. Thus, $seg_{i,1}$ has a spanning direction from left to right.

Because a microstrip segment can take only one direction, directional variables satisfy the following constraint:

$$s^u_{i,j}+s^d_{i,j}+s^l_{i,j}+s^r_{i,j}=1,\ \forall i\le m,\ \forall j\le n_i-1. \tag{1}$$

Since the chain point c_{i,n_i} of the i^{th} microstrip denotes the end of this microstrip, no segment spans from c_{i,n_i} and thus (1) is not applied to c_{i,n_i}.

To prevent that the $(j+1)^{th}$ segment goes back to the j^{th} chain point, the variables representing two reversed directions at two consecutive chain points cannot be 1 at the same time, hence

$$s^l_{i,j}+s^r_{i,j+1}\le 1, \tag{2}$$
$$s^d_{i,j}+s^u_{i,j+1}\le 1, \tag{3}$$
$$s^r_{i,j}+s^l_{i,j+1}\le 1, \tag{4}$$
$$s^u_{i,j}+s^d_{i,j+1}\le 1. \tag{5}$$

2) Spanning length

To model the length $l_{i,j}$ of the microstrip segment $seg_{i,j}$, we introduce four more integer variables $l^l_{i,j}$, $l^d_{i,j}$, $l^r_{i,j}$, and $l^u_{i,j}$ to represent the spanning lengths in four directions and

$$l_{i,j}=l^l_{i,j}+l^d_{i,j}+l^r_{i,j}+l^u_{i,j}. \tag{6}$$

Only if a segment is spanned to a certain direction, the value of the corresponding spanning length can be other than 0. In addition, the spanning length should be larger than or equal to the minimum segment length κ specified by the design rules. We thus model the relation between spanning directions and spanning lengths as

$$s^u_{i,j}\kappa\le l^u_{i,j}\le \mathcal{M}s^u_{i,j}, \tag{7}$$
$$s^d_{i,j}\kappa\le l^d_{i,j}\le \mathcal{M}s^d_{i,j}, \tag{8}$$
$$s^l_{i,j}\kappa\le l^l_{i,j}\le \mathcal{M}s^l_{i,j}, \tag{9}$$
$$s^r_{i,j}\kappa\le l^r_{i,j}\le \mathcal{M}s^r_{i,j}, \tag{10}$$

where $\mathcal{M}$ is a very large constant. If a binary directional variable is set to 1, the right part of the inequation will become trivial, and the left part of the inequation will force the corresponding spanning length to be larger than or equal to κ. However, if the binary directional variable is set to 0, the right part of the inequation will force the corresponding spanning length to be 0.

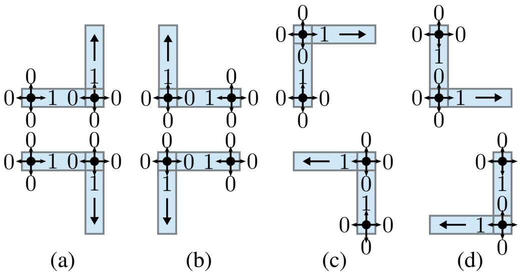


Figure 5: All kinds of bends described using the directional variables at chain points.

With the spanning length variables, the relation between any two consecutive chain points $c_{i,j}$, $c_{i,j+1}$, $j\in\{1,\cdots,n_i-1\}$ can be described by

$$x_{i,j+1}=x_{i,j}-l^l_{i,j}+l^r_{i,j}, \tag{11}$$
$$y_{i,j+1}=y_{i,j}-l^d_{i,j}+l^u_{i,j}. \tag{12}$$

Moreover, the geometrical length of a microstrip line (without considering the length compensation owing to introducing bends) can thus be calculated by summing up the lengths of all segments,

$$l_{g,i}=\sum_{j=1,\ldots n_i-1} l_{i,j},\ \forall i\le m. \tag{13}$$

B. Modelling Bends with Chain Points

As discussed in Section II-B, the discontinuity effect exists at each bend on the microstrip. A bend happens when two consecutive segments take different directions to span, as shown in Figure 4(c). To describe the condition whether a bend is really formed at a chain point in the model, we assign a new binary variable $t_{i,j}$ to represent bend existence for the chain point at $(x_{i,j}, y_{i,j})$. $t_{i,j}$ is set to 1 if $seg_{i,j}$ takes a direction different from $seg_{i,j-1}$, and thus a bend exists at $(x_{i,j}, y_{i,j})$.

The situations under which a bend is created are summarized in Figure 5. Only one of these situations may happen at a chain point if there is a bend, so the condition for the presence of a bend can be described as

$$s^r_{i,j-1}+s^l_{i,j-1}+s^u_{i,j}+s^d_{i,j}=2t_{i,j,hv}+u_{i,j,hv}, \tag{14}$$
$$s^u_{i,j-1}+s^d_{i,j-1}+s^r_{i,j}+s^l_{i,j}=2t_{i,j,vh}+u_{i,j,vh}, \tag{15}$$
$$t_{i,j}=t_{i,j,hv}+t_{i,j,vh}\le 1, \tag{16}$$

where $t_{i,j,hv}$, $u_{i,j,hv}$, $t_{i,j,vh}$ and $u_{i,j,vh}$ are auxiliary binary variables. (14) is the constraint for the cases in Figures 5(a) and (b), where $s^r_{i,j-1}$ and $s^l_{i,j-1}$ cannot be 1 at the same time according to (1), and neither can $s^u_{i,j}$ and $s^d_{i,j}$. If any of the four situations in Figures 5(a) and (b) happens, the sum on the left side of (14) is equal to 2 so that $t_{i,j,hv}$ must be set to 1. Otherwise, $t_{i,j,hv}$ must be set to 0. Similar to (14), (15) is the constraint for the cases in Figures 5(c) and (d). Combining these situations together, a bend is created when either $t_{i,j,hv}=1$ or $t_{i,j,vh}=1$. Therefore, the variable $t_{i,j}$ representing the presence of a bend can be constrained by (16), where $t_{i,j,hv}$ and $t_{i,j,vh}$ cannot be 1 at the same time.

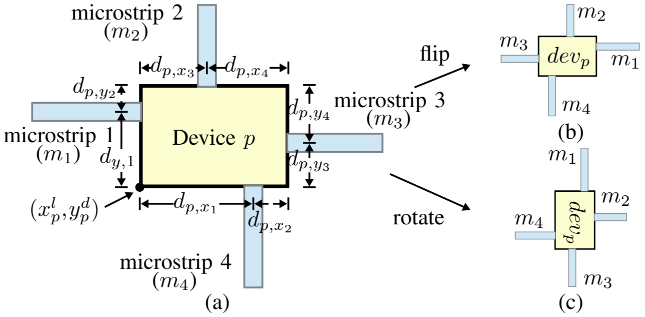


Figure 6: Different pin locations by applying different orientation changing rules : (a) A four-pin device model. (b) Pin location changes after flipping. (c) Pin location changes after rotating.

The total number of real bends formed on the i^{th} microstrip can thus be described as

$$n_{b,i}=\sum_{j=2,\dots n_i-1} t_{i,j},\ \forall i\leq m. \tag{17}$$

According to Section II-B, a 90° bend will be replaced by a diagonal pattern in the final routing, as illustrated in Figure 3. For each bend, we need to compensate the length of the microstrip by δ. Combining with the binary bending variable $t_{i,j}$ with the geometrical length of a microstrip defined in (13), we can write the equivalent length of the i^{th} microstrip as

$$l_{eq,i}=l_{g,i}+\sum_{j=2,\dots n_i-1} t_{i,j}\delta. \tag{18}$$

This equivalent length must be equal to the target length, L_i, of the i^{th} microstrip according to the specification. Consequently,

$$l_{eq,i}=L_i. \tag{19}$$

Note that δ in (18) is usually negative, and thus the geometrical length of a microstrip is required to be longer than the equivalent length, which means that the microstrip line needs to be routed longer at the end if it contains more bends.

C. Modelling Connections to Devices

The two ends of a microstrip should be connected to devices. Assume that the i^{th} microstrip is connected with a device p at $(x_{i,j}, y_{i,j})$, which is equal to $(x_{i,1}, y_{i,1})$ when the first chain point of the microstrip is connected, or (x_{i,n_i}, y_{i,n_i}) when the last chain point of the microstrip is connected. We further assume that the coordinate of the lower-left corner of the device p is (x_p^l, y_p^d), and the connection pin of the deivce and the i^{th} microstrip has the offset $(o_{p,i,x}, o_{p,i,y})$ from the lower-left corner. As this pin is connected to the chain point $(x_{i,j}, y_{i,j})$, the two coordinates must be the same, which can be described as

$$x_{i,j}=x_p^l+o_{p,i,x} \text{ and } y_{i,j}=y_p^d+o_{p,i,y}. \tag{20}$$

On some devices, the pin connection can be switched. In addition, devices may be able to be flipped or rotated, which can lead to changes of pin locations. For example, in Figure 6(a), a four-pin device p is shown. The pin location connected with the left microstrip line m_1 is $(x_p^l, y_p^d+d_{p,y_1})$ ($o_{p,i,x}=0$ and $o_{p,i,y}=d_{p,y_1}$). If this device is flipped or rotated as shown in Figure 6(b)(c), the pin location connected to m_1 becomes $(x_p^l+d_{p,x_1}+d_{p,x_2}, y_p^d+d_{p,y_1})$ when flipped, or $(x_p^l+d_{p,y_1}, y_p^d+d_{p,x_1}+d_{p,x_2})$ when rotated.

To choose the best device orientation and pin locations after considering device flipping and rotating as well as pin-location switching, we assign a binary variable e for connection combination selection. The k^{th} combination is chosen if and only if e_k is set to 1, and only one e_k among all e is 1. This constraint can be written by transforming (20) into

$$x_{i,j}=x_p^l+\sum_{k=1,\dots\epsilon} e_k o_{p,i,x,k}, \tag{21}$$

$$y_{i,j}=y_p^d+\sum_{k=1,\dots\epsilon} e_k o_{p,i,y,k}, \tag{22}$$

$$\sum_{k=1,\dots\epsilon} e_k=1, \tag{23}$$

where ϵ is a constant representing the number of available combinations, $o_{p,i,x,k}$ is the k^{th} $o_{p,i,x}$, and $o_{p,i,y,k}$ is the k^{th} $o_{p,i,y}$. Suppose that there are three combinations ($\epsilon=3$) in the example shown in Figure 6, and each combination is shown in Figure 6(a), (b), and (c) respectively, With (21)–(23), the x-axis coordinate of the pin location connected with m_1 can be denoted by $x_p^l+e_2(d_{p,x_1}+d_{p,x_2})+e_3 d_{p,y_1}$, and the y-axis coordinate can be denoted by $y_p^d+(e_1+e_2)d_{p,y_1}+e_3(d_{p,x_1}+d_{p,x_2}))$.

Pads are a special type of devices. Unlike other devices, every location surrounding a pad can be treated as an available pin location. (21)–(23) are thus no longer applicable for modelling the connections between microstrips and pads. We modify the constraints as follows:

$$x_{i,j}\geq x_p^l+e_1 d_{p,h}, \tag{24}$$

$$x_{i,j}\leq x_p^l+d_{p,h}-e_2 d_{p,h}, \tag{25}$$

$$y_{i,j}\geq y_p^d+e_3 d_{p,v}, \tag{26}$$

$$y_{i,j}\leq y_p^d+d_{p,v}-e_4 d_{p,v} \tag{27}$$

$$e_1+e_2+e_3+e_4=1, \tag{28}$$

where $d_{p,h}$ and $d_{p,v}$ are constants representing the horizontal and vertical dimensions of pad p. With (24)-(28), for example, if e_1 is set to 1 and e_2–e_4 are set to 0 according to (28), $x_{i,j}$ must be $x_p^l+d_{p,h}$ according to (24) and (25), as $y_{i,j}$ is still free to be chosen from y_p^d to $y_p^d+d_{p,v}$ according to (26) and (27).

Note that all pad types in our device library are either square or have fixed orientations, thus pad flip or rotation do not need to be considered in our method. As a pad can be connected to a microstrip from anywhere surrrounding the pad, pin-location switching does not need to be considered either. Therefore, we do not assign combination-selection variables to pads. This specification can easily be extended, if a new pad type that does not follow the connection rules for other pads is introduced. In this case, we simply need to put constraints (21)–(23) back on those new pads again.

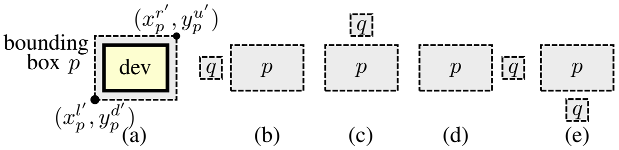


Figure 7: (a) Corner coordinates of bounding box p. (b)–(e) Non-overlapping situations.

D. Modelling Non-Overlapping Bounding Boxes

To guarantee layout planarity, microstrip lines and devices must not overlap with each other. In addition, the spacing distance among microstrip lines and devices must satisfy the design rule.

In our method, we propose a bounding box concept to model the spacing distance specification of microstrip lines and devices. As mentioned in Section II-A, a bounding box is expanded from either a microstrip segment or a device. If a bounding box contains a microstrip segment, it expands both the horizontal and vertical dimension of the corresponding segments by a constant value t; if a bounding box is expanded from a device, the expansion depends on the design rule on the corresponding device so that the minimum spacing distance can be ensured. Therefore, as long as two bounding boxes do not overlap with each other, the spacing distance between microstrips and devices is automatically guaranteed.

We then introduce non-overlapping constraints to our model to ensure the layout planarity. Assume that there are two bounding boxes p and q, the coordinates of the lower-left corner and the upper-right corner of the two bounding boxes are $(x_p^{l'}, y_p^{d'})$, $(x_p^{r'}, y_p^{u'})$; and $(x_q^{l'}, y_q^{d'})$, $(x_q^{r'}, y_q^{u'})$ respectively, as shown in Figure 7(a). To prevent the two bounding boxes from overlapping with each other, their relative locations must be one of the situations in Figures 7(b)–(e), and the corner coordinates of the bounding boxes should meet

$$x_q^{r'} \le x_p^{l'} + \mathcal{M} q_1, \tag{29}$$

$$y_p^{u'} \le y_q^{d'} + \mathcal{M} q_2, \tag{30}$$

$$x_p^{r'} \le x_q^{l'} + \mathcal{M} q_3, \tag{31}$$

$$y_q^{u'} \le y_p^{d'} + \mathcal{M} q_4, \tag{32}$$

$$q_1 + q_2 + q_3 + q_4 = 3, \tag{33}$$

where q_1, q_2, q_3, and q_4 are auxiliary binary variables, and $\mathcal{M}$ again is a very large constant. The constraint (33) requests that one variable from q_1–q_4 to be set to 0, and hence at least one of the four situations from Figure 7(b)–(e) is guaranteed.

E. Pads on Chip Boundary

Pads are a special type of devices that are required to be placed along the chip boundary for inter-chip communication. To allocate a pad p to either the upper, lower, left, or right boundary, we apply the following constraints:

$$x_p^{l'} \le \rho_1 + \mathcal{M} q_1, \tag{34}$$

$$x_p^{r'} \ge L_h - \rho_2 - \mathcal{M} q_2, \tag{35}$$

$$y_p^{d'} \le \rho_3 + \mathcal{M} q_3, \tag{36}$$

$$y_p^{u'} \ge L_v - \rho_4 - \mathcal{M} q_4, \tag{37}$$

$$q_1 + q_2 + q_3 + q_4 = 3, \tag{38}$$

where ρ_1, ρ_2, ρ_3, and ρ_4 are constants, which can be customized to control the maximum distance between the pad boundary and the chip boundary, $\mathcal{M}$ again is a very large constant, L_h and L_v represent the chip dimensions, and q_1, q_2, q_3, and q_4 are auxiliary binary variables. With (38), one of q_1, q_2, q_3 and q_4 must be set to 0, and its corresponding constraint is thus activated, while other constraints become trivial. The activated constraint indicates the chip boundary, along which p is located. if q_1 is set to 0, q_2, q_3, and q_4 are automatically set to 1. Thus, constraints (35)–(37) become trivial, and constraint (34) ensures that pad p is located along the left chip boundary.

F. Chip Area Confinement

Since microstrips and devices must not be located beyond the chip area, we confine the coordinates of bounding boxes by

$$x_p^{r'} <= L_h \text{ and } y_p^{u'} <= L_v, \ \forall p \le |B|, \tag{39}$$

where B is a set containing all bounding boxes.

G. Optimization Formulation

We define the complete ILP model for exact placement and routing of an RFIC as

$$\text{Minimize:} \quad \alpha n_{b,max} + \beta \sum_{i=1,\ldots m} n_{b,i} \tag{40}$$

$$\text{Subject to: } (1)\text{–}(19), (21)\text{–}(39), \tag{41}$$

where $n_{b,i}$ is defined in (17) and indicates the number of bends on the i^{th} microstrip, $n_{b,max}$ is the maximal number of $n_{b,i}$, $i \in \{1, \cdots, m\}$, and α and β are adjustable weight coefficients that can be defined by the designer.

To solve this optimization problem, the ILP solver needs to search the whole chip area to determine the location of each device and each chain point, which results in very heavy calculation burden.

V. Progressive ILP-based RFIC Layout Generation

Although the complete ILP model proposed in the previous section can generate an exact layout, including placement and routing solutions, the runtime for solving this model is not acceptable. In order to generate the chip layout more efficiently, we propose a improved progressive ILP-based (IP-ILP) method consisting of two phases. In the first phase, we omit some detailed constraints such as the constraints on device orientation, pin location, and spanning direction of microstrips, and keep the high-impact constraints such as the constraints on layout planarity and device connection relations in the model. The solution of this first phase model is a global chip layout, which is then iteratively refined and validated in the second phase. With our progressive modelling method, the calculation burden of this problem is greatly reduced, and thus an optimized RFIC layout can be automatically generated within reasonable runtime.

A. Global Layout Generation

In the first modelling phase, we generate a *global layout* indicating suitable locations for both microstrips and devices.

1) Microstrip model

Different from the complete model, in which microstrips are divided by chain points into several segments, we omit the inner chain points of microstrips in our global layout generation phase, and thus all microstrips are modelled as straight lines without bends.

We denote the coordinates of the endpoints of the i^{th} microstrip as $(x_{i,0},y_{i,0})$ and $(x_{i,1},y_{i,1})$, and represent the length of a microstrip as the Manhattan distance between its endpoints. We denote the length of the i^{th} microstrip as l_i, and thus

$$l_i=|x_{i,0}-x_{i,1}|+|y_{i,0}-y_{i,1}|. \tag{42}$$

We linearize this equation to integrate it into our integer-linear-programming model:

$$l_i>=(x_{i,0}-x_{i,1})+(y_{i,0}-y_{i,1}), \tag{43}$$
$$l_i>=(x_{i,1}-x_{i,0})+(y_{i,0}-y_{i,1}), \tag{44}$$
$$l_i>=(x_{i,0}-x_{i,1})+(y_{i,1}-y_{i,0}), \tag{45}$$
$$l_i>=(x_{i,1}-x_{i,0})+(y_{i,1}-y_{i,0}), \tag{46}$$
$$l_i<=(x_{i,0}-x_{i,1})+(y_{i,0}-y_{i,1})+\mathcal{M}q_1, \tag{47}$$
$$l_i<=(x_{i,1}-x_{i,0})+(y_{i,0}-y_{i,1})+\mathcal{M}q_2, \tag{48}$$
$$l_i<=(x_{i,0}-x_{i,1})+(y_{i,1}-y_{i,0})+\mathcal{M}q_3, \tag{49}$$
$$l_i<=(x_{i,1}-x_{i,0})+(y_{i,1}-y_{i,0})+\mathcal{M}q_4, \tag{50}$$
$$q_1+q_2+q_3+q_4=3, \tag{51}$$

where q_1, q_2, q_3, and q_4 are auxiliary binary variables and $\mathcal{M}$ is a very large constant. (43)–(46) ensure that $l_i \geq \max\{(x_{i,0}-x_{i,1})+(y_{i,0}-y_{i,1}), (x_{i,1}-x_{i,0})+(y_{i,0}-y_{i,1}), (x_{i,0}-x_{i,1})+(y_{i,1}-y_{i,0}), (x_{i,1}-x_{i,0})+(y_{i,1}-y_{i,0})\}$. (47)–(51) ensure that $\exists \iota \in \{(x_{i,0}-x_{i,1})+(y_{i,0}-y_{i,1}), (x_{i,1}-x_{i,0})+(y_{i,0}-y_{i,1}), (x_{i,0}-x_{i,1})+(y_{i,1}-y_{i,0}), (x_{i,1}-x_{i,0})+(y_{i,1}-y_{i,0})\}$, $l_i \leq \iota$. Thus, we can conclude that $l_i=\iota=\max\{(x_{i,0}-x_{i,1})+(y_{i,0}-y_{i,1}), (x_{i,1}-x_{i,0})+(y_{i,0}-y_{i,1}), (x_{i,0}-x_{i,1})+(y_{i,1}-y_{i,0}), (x_{i,1}-x_{i,0})+(y_{i,1}-y_{i,0})\}$, which represents the Manhattan distance.

In our global layout generation phase, instead of connecting microstrips to pins at device boundaries, we assume that microstrips are connected to the center points of devices (details of device connection constraints will be discussed in Section V-A2). In order to reduce the influence of device dimension on microstrip length, we introduce a new integer variable $l_{eq,i}$ to approximate the real length of the i^{th} microstrip. Suppose that the endpoints of the i^{th} microstrip are connected to device a and device b, d_a represents the longest side length of device a, and d_b represents the longest side length of device b. We apply the following constraint to set the value of $l_{eq,i}$:

$$l_{eq,i}=l_i-\frac{d_a}{2}-\frac{d_b}{2}. \tag{52}$$

Our microstrip model in the global layout generation phase is a simplified version of the complete microstrip model mentioned in Section IV-A. Since $l_{eq,i}$ only approximates but does not represent the real length of the i^{th} microstrip in the final layout, we do not force $l_{eq,i}$ to meet the pre-defined target microstrip length L_i. Instead, we perform a Lagrangian-relaxation by introducing two integer variables $l_{abs,i}$ and $l_{rel,i}$ to model the mismatch between $l_{eq,i}$ and L_i, and set the minimization of microstrip length mismatch as our optimazition objective. $l_{abs,i}$ represents the absolute mismatch between $l_{eq,i}$ and L_i, and is thus calculated as $l_{abs,i}=|L_i-l_{eq,i}|$, which is linearized by the following constraints:

$$l_{abs,i}\geq L_i-l_{eq,i}, \tag{53}$$
$$l_{abs,i}\geq -(L_i-l_{eq,i}). \tag{54}$$

Although (53) and (54) only suggest that $l_{abs,i}\geq|L_i-l_{eq,i}|$, with the minimization of the absolute length mismatch as our modelling objective, we can achieve the minimum acceptable $l_{abs,i}$ automatically. $l_{rel,i}$ represents the relative mismatch between $l_{eq,i}$ and L_i, which is calculated as $l_{rel,i}=\frac{l_{abs,i}}{L_i}$, and helps to decide the optimization priority. For example, assume that there are two microstrips i_1 and i_2 with $l_{eq,i_1}=30\mu m$ and $l_{eq,i_2}=210\mu m$, and the target length of i_1 and i_2 are $L_{i_1}=20\mu m$ and $L_{i_2}=200\mu m$. Although the absolute length mismatch of the two microstrips can be calculated as $l_{abs,i_1}=l_{abs,i_2}=10\mu m$, the relative mismatch of i_1 is much more serious than i_2, since $l_{rel,i_1}=\frac{10}{20}=0.5>0.05=l_{rel,i_2}$. Therefore, the mismatch of the i_1^{th} microstrip should be optimized prior to the mismatch of the i_2^{th} microstrip.

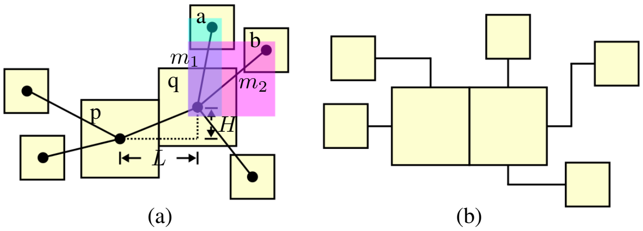
2) Modelling connections to devices

In the complete model, the model for device connections results in a calculation burden, since tens of devices of different dimensions are modelled with changeable orientations and pin orders. Therefore, the solution space becomes so large that the computation may not be affordable.

In our global layout generation phase, we model devices (except for pads) as squares instead of rectangles. We choose the longest side length of a device as the side length of the new square, and thus devices of different orientations can be restored in the later modelling phase without area concern. In addition, we omit the pins of devices in this phase. Instead of connecting microstrips to pins at device boundaries, we assume that microstrips are connected to the center points of devices including pads.

We denote the coordinates of the center point of device p as (x^c_p,y^c_p). Assume that $(x_{i,j},y_{i,j})$ is the coordinate of a microstrip endpoint that is connected to p. Then the following constraints must be fulfilled:

$$x^c_p=x_{i,j}, \tag{55}$$
$$y^c_p=y_{i,j}. \tag{56}$$



(a) (b)

Figure 8: (a) An example of a global layout. (b) A possible final layout based on (a).

Therefore, the detailed orientation- and pin-selection model described in constraints (21)–(28) can be omited, and calculation efforts are saved.

For devices that are directly connected with each other without microstrips, we control the distance between them to ensure that they are adjacent to each other. Assume that device p is directly connected with another device q. Then we denote the horizontal and vertical dimension of p and q as $d_{p,h}$, $d_{p,v}$, and $d_{q,h}$, $d_{q,v}$, respectively. Note that if p (or q) is not a pad, then $d_{p,h} = d_{p,v}$ (or $d_{q,h} = d_{q,v}$). We introduce the following constraints to ensure that the horizontal (and vertical) distance between the two center points of p and q are shorter than or equal to the summation of the half horizontal (and vertical) dimensions of p and q:

$$x_p^c - x_q^c \geq -(\frac{d_{p,h}}{2} + \frac{d_{q,h}}{2}), \tag{57}$$

$$x_p^c - x_q^c \leq \frac{d_{p,h}}{2} + \frac{d_{q,h}}{2}, \tag{58}$$

$$y_p^c - y_q^c \geq -(\frac{d_{p,v}}{2} + \frac{d_{q,v}}{2}), \tag{59}$$

$$y_p^c - y_q^c \leq \frac{d_{p,v}}{2} + \frac{d_{q,v}}{2}. \tag{60}$$

Since devices are prevented from overlapping with each other (details of the non-overlapping constraints are discussed in Section V-A3), above constraints ensure that p and q are placed adjacent to each other.

An example of device connection is shown in Figure 8(a), where microstrips are connected to the center points of devices, and devices that are connected without microstrips are placed adjacent. L represents the horizontal distance of the center points of device p and device q, and H represents the vertical distance of the center points. Since the overlap of p and q are prohibited, either L or H must be equal to the summation of the half horizontal or vertical side length of p and q, and thus the connection is ensured. Based on the global layout, device dimensions, orientations, and pin locations will be restored in the later validation phase, Figure 8(b) shows a possible final layout restored from Figure 8(a) (Details of the validation phase will be discussed in Section V-B).

3) Modelling non-overlapping conditions

The above mentioned simplification of the microstrip model and the device connection model enables us to reduce the number of non-overlapping constraints.

Similar as in the complete model, devices and microstrips are also expanded as bounding boxes in the global layout generation phase. However, different from the complete model, where microstrips are connected to the pins at the device boundaries, we connect microstrips to center points of devices, which results in two classes of necessary overlap among bounding boxes: for a device and a microstrip connected to this device, their bounding boxes necessarily overlap with each other, since the microstrip goes over the device boundary; and for two microstrips that are connected to the same device, their bounding boxes also necessarily overlap with each other, since they share the same endpoints. For bounding boxes beyond these two classes, we inherit constraints (29)–(33) from the complete model to prohibit the undesirable overlapping.

The application of non-overlapping constraints can be illustrated by Figure 8(a), where m_1 and m_2 are two microstrips that are both connected to the center point of device q. In addition, the other endpoint of m_1 is connected to device a, and the other endpoint of m_2 is connected to device b. Although we do not apply non-overlapping constraints to the bounding boxes expanded from m_1, m_2, and q, a feasible solution can still be generated, since overlapping among different devices and among microstrips without shared endpoints is prohibited.

4) Opposite neighbor devices

According to the device specifications, most 2-pin devices have their pins located at opposite device boundaries, and most 4-pin devices have two pairs of pins, each of which has two equivalent pins located at opposite device boundaries. In our work, if two devices a and b are connected to the pins located at opposite boundaries of a device p, we call a and b as the *opposite neighbor devices* of p, and p as the *reference device* of a and b.

In our global layout generation phase, although we omit the pins and assume that devices are connected at their center points, we still take the specifications of opposite neighbor devices into consideration to prevent a potential net order problem.

For example, suppose that $\{a,b\}$ and $\{c,d\}$ are two pairs of opposite neighbor devices of p. Without considering this specification, a and b could be placed at the same side of p as shown in Figure 9(a), which seems reasonable in the global layout generation phase, but results in a long detour of microstrip routing, when we restore the pin locations and connect microstrips back to pins in the later phase. This long detour leads to various problems including area competition, length matching violation, etc., which requires a remarkable change of the whole layout. In order to reduce the refinement effort, we rule that opposite neighbor devices must be placed at the opposite sides of their reference device. An exemplary layout fulfilling this new rule is shown in Figure 9(b), which is easy to be refined for pin restoration.

For a device p that has one pair of opposite neighbor devices $\{a,b\}$, the new placement rule can be integrated into our model

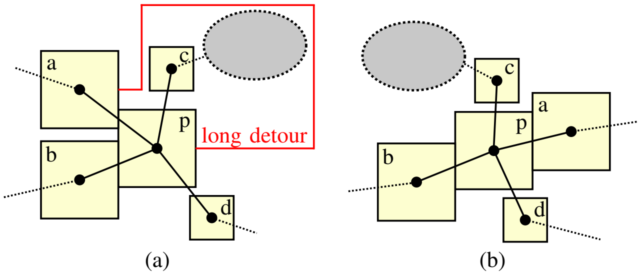


Figure 9: Two possible global layout results. (a) Without considering opposite neighbor devices. (b) Considering opposite neighbor devices.

by introducing the following constraints:

$$x_a^r \le x_p^l + \mathcal{M}q_1 \ \wedge \ x_b^l \ge x_p^r - \mathcal{M}q_1, \tag{61}$$

$$x_a^u \le x_p^d + \mathcal{M}q_2 \ \wedge \ x_b^d \ge x_p^u - \mathcal{M}q_2, \tag{62}$$

$$x_b^r \le x_p^l + \mathcal{M}q_3 \ \wedge \ x_a^l \ge x_p^r - \mathcal{M}q_3, \tag{63}$$

$$x_b^u \le x_p^d + \mathcal{M}q_4 \ \wedge \ x_a^d \ge x_p^u - \mathcal{M}q_4, \tag{64}$$

$$q_1 + q_2 + q_3 + q_4 = 3. \tag{65}$$

With (65), one constraint among (61)–(64) will be activated by setting corresponding q to 0, and thus device a and device b will be placed in opposite directions regarding device p.

For a 4-pin device p that has two pairs of opposite neighbor devices $\{a,b\}$ and $\{c,d\}$, we extend constraints (61)–(65), by introducing the following constraints:

$$x_a^r \le x_p^l + \mathcal{M}q_5 \ \wedge \ x_b^l \ge x_p^r - \mathcal{M}q_5, \tag{66}$$

$$x_a^u \le x_p^d + \mathcal{M}q_6 \ \wedge \ x_b^d \ge x_p^u - \mathcal{M}q_6, \tag{67}$$

$$x_b^r \le x_p^l + \mathcal{M}q_7 \ \wedge \ x_a^l \ge x_p^r - \mathcal{M}q_7, \tag{68}$$

$$x_b^u \le x_p^d + \mathcal{M}q_8 \ \wedge \ x_a^d \ge x_p^u - \mathcal{M}q_8, \tag{69}$$

$$q_5 + q_6 + q_7 + q_8 = 3. \tag{70}$$

In addition, we further introduce the following constraint to make sure that one pair of p's opposite neighbor devices must be placed in the horizontal direction opposite to p:

$$q_1 + q_3 + q_5 + q_7 = 3, \tag{71}$$

and with (65), (70), and (71), $q_2 + q_4 + q_6 + q_8$ is automatically equal to 3, which means the other pair of p's opposite neighbor devices must be placed in the vertical direction opposite to p.

5) Optimization formulation

As mentioned in Section V-A1, we omit the bends of microstrips in the global layout generation phase. Therefore, instead of optimizing the number of bends as in our complete model, we optimize the length mismatch of microstrips to improve the accuracy of our model.

As mentioned in Section V-A1, we denote the absolute and relative length mismatch of the i^{th} microstrip as $l_{abs,i}$ and $l_{rel,i}$ respectively. Now we define our optimization objective of the global model as minimizing the total absolute length mismatch, as well as the maximal relative length mismatch. Constraints (34)–(38) modelling pad locations and constraint (39) modelling chip area confinement are inherited from the complete model. Therefore, the optimization model for global layout generation can be formulated as:

$$\text{Minimize:} \quad \zeta \sum_{i=1,\ldots m} l_{abs,i} + \gamma \max_{i=1,\cdots,m} l_{rel,i} \tag{72}$$

$$\text{Subject to: } (29)\text{–}(39), (43)\text{–}(71), \tag{73}$$

where ζ and γ are adjustable weight coefficients that can be defined by the user.

B. Iterative Validation

In the global layout generation phase, microstrips are modelled as straight lines without bends, and are connected to the center points of devices in an any-angled manner. Thus, the coordinates of microstrip endpoints not only represent the routing solutions of microstrips, but also represent the locations of devices. In other word, the global layout is characterized by the coordinates of microstrips endpoints.

Since we omit the bends of microstrips in the global model, current device locations need to be modified to generate the final layout that meets all design rules. Therefore, we inherit the endpoints information from the global model, and restrict the variation range of device locations in the later *iterative layout validation phase*.

The model reduction in the global layout generation phase results in three possible violations of design rules that need to be satisfied in the final design: 1. The length of microstrips must be equal to the pre-defined target length, 2. Microstrips should be connected to pins at device boundaries, instead of center points, 3. Microstrips should be routed only in horizontal or in vertical direction, instead of any-angled routing.

In order to eliminate these violations, we add additional chain points of microstrips to our model, and restore the routing directions and pin connections progressively in several iterations. Instead of prohibiting the violations of design rules, we tolerate them by relaxing some of the constraints describing design rules, but penalize these violations by adding extra costs in the optimization objective. The opimization process is performed iteratively in this phase until a solution satisfying all design rules is obtained.

Many constraints of the layout validation model can be inherited from the complete model, including the constraints describing microstrip bends, pad locations and chip area confinement. On the other hand, constraints on microstrip lengths, microstrip bounding box dimension, minimum microstrip segment length, microstrip spanning direction, and pin location of devices, are relaxed. Detailed relaxation rules are described in the rest of this section.

1) Bounding box relaxation of microstrips

As mentioned earlier, bounding boxes of microstrips and devices are prohibited from overlapping with one another to avoid the coupling effect. In order to make the non-overlapping constraints easier to be satisfied, we relax the constraints on bounding box dimensions of microstrips, so that the dimensions of microstrip bounding boxes can be smaller than the dimensions specified in the complete model.

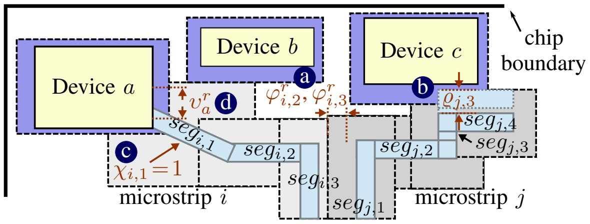


Figure 10: An example of relaxations. (a) Bounding box area(φ). (b) Minimum segment length(ϱ). (c) Spanning direction(χ) (d) Pin location(υ).

This is realized by transforming constraints (29)–(33) into the following:

$$x_q^{r'} - \varphi_q^{r'} \leq x_p^{l'} + \varphi_p^{l'} + \mathcal{M}q_1, \quad (74)$$

$$y_p^{u'} - \varphi_p^{u'} \leq y_q^{d'} + \varphi_q^{d'} + \mathcal{M}q_2, \quad (75)$$

$$x_p^{r'} - \varphi_p^{r'} \leq x_q^{l'} + \varphi_q^{l'} + \mathcal{M}q_3, \quad (76)$$

$$y_q^{u'} - \varphi_q^{u'} \leq y_p^{d'} + \varphi_p^{d'} + \mathcal{M}q_4, \quad (77)$$

$$q_1 + q_2 + q_3 + q_4 = 3, \quad (78)$$

where p can also be denoted as (i,m), if bounding box p contains the m^{th} segment of the i^{th} microstrip, and φ is a variable representing the dimension decrease of the corresponding bounding box. The lower bound of φ is constantly set to 0, and the initial upper bound of φ is set as $\frac{w}{2}+t$, where w is the width of microstrips specified in the design rules, and t is half of the minimum spacing distance between two microstrips as mentioned in Section II-A. We add the minimization of φ to our optimization objective for constraint relaxation, and update the upper bound of φ as the value of φ generated in the last iteration.

An example is shown in Figure 10(a), where the i^{th} and the j^{th} microstrips are suffering from routing area competition. With the relaxation of the constraints on microstrip bounding boxes, the bounding box of the i^{th} microstrip is allowed to be shrunk, and thus the two bounding boxes are routed within limited space to generate an intermediate result.

2) Minimum microstrip segment length relaxation

For each microstrip segment, there is a minimum segment length d_{min} specified in the design rules. We relax this rule by allowing the minimum segment length of the j^{th} segment of the i^{th} microstrip to be shortened by $\varrho_{i,j}$, which has an initial upper bound d_{up} that fulfills $d_{up} < d_{\text{min}}$, so that the minimum segment length cannot be decreased to 0. We add the minimization of ϱ to our optimization objective for constraint relaxation, and update the upper bound of ϱ as the value of ϱ generated in the last iteration.

An example is shown in Figure 10(b), where device c competes for chip area with the 3^{rd} segment of the j^{th} microstrip, denoted as $seg_{j,3}$, and the length of $seg_{j,3}$ is equal to the minimum segment length specified in design rules. Since we do not permit the relaxation of bounding box dimensions of devices, and the dimension of the bounding box of $seg_{j,3}$ has already shrunk to the minimum, the minimum segment length of $seg_{j,3}$ is shortened, so that an intermediate result can be generated within limited space.

3) Microstrip spanning direction relaxation

According to design rules, microstrips are routed only in horizontal or in vertical direction. This rule is described in the complete model by introducing four binary directional variables: $s_{i,j}^l$, $s_{i,j}^d$, $s_{i,j}^r$, and $s_{i,j}^u$, exactly one of which must be set to 1, and thus ensures that the i^{th} microstrip can only span in one horizontal or vertical direction.

In our iterative layout validation model, we relax this rule by introducing a new binary variable χ, thus transforming the direction constraint (1) into the following:

$$s_{i,j}^u + s_{i,j}^d + s_{i,j}^l + s_{i,j}^r = 1 + \chi_{i,j}, \; \forall i \leq m, \; \forall j \leq n_i - 1. \quad (79)$$

If $\chi_{i,j}$ is set to 1, above constraint means that two of the directional variables will be set to 1, too, and thus the i^{th} microstrip is allowed to be routed in an any-angled manner. We add the minimization of χ to our optimization objective to penalize this violation. In order to prevent that variables representing opposite directions are chosen together, we also introduce the following constraints:

$$s_{i,j}^l + s_{i,j}^r \leq 1 \quad (80)$$

$$s_{i,j}^d + s_{i,j}^u \leq 1. \quad (81)$$

An example is shown in Figure 10(c), where $\chi_{i,1}$ is set to 1, so that $seg_{i,1}$ is allowed to be routed as an oblique line, eg. for the sake of length matching.

4) Pin location relaxation

According to design rules, pins of devices are located at fixed positions at device boundaries. We relax this rule by allowing but penalizing the deviation of pin locations along the boundaries of devices. We represent the deviation of the location of a pin by introducing four binary variables υ_p^l, υ_p^d, υ_p^r, and υ_p^u, where p indicates the device that the pin belongs to, and $\{l,d,r,u\}$ represent the four directions. We then transform constraints (21) and (22) into the following:

$$x_{i,j} \leq x^l + \sum_{k=1,\dots\epsilon} e_k x_{off,i,k} + \upsilon_p^d + \mathcal{M}(1 - \sum_{k \in P^d} e_k), \quad (82)$$

$$x_{i,j} \geq x^l + \sum_{k=1,\dots\epsilon} e_k x_{off,i,k} + \upsilon_p^d - \mathcal{M}(1 - \sum_{k \in P^d} e_k), \quad (83)$$

$$x_{i,j} \leq x^l + \sum_{k=1,\dots\epsilon} e_k x_{off,i,k} + \upsilon_p^u + \mathcal{M}(1 - \sum_{k \in P^u} e_k), \quad (84)$$

$$x_{i,j} \geq x^l + \sum_{k=1,\dots\epsilon} e_k x_{off,i,k} + \upsilon_p^u - \mathcal{M}(1 - \sum_{k \in P^u} e_k), \quad (85)$$

$$y_{i,j} \leq y^d + \sum_{k=1,\dots\epsilon} e_k y_{off,i,k} + \upsilon_p^l + \mathcal{M}(1 - \sum_{k \in P^l} e_k), \quad (86)$$

$$y_{i,j} \geq y^d + \sum_{k=1,\dots\epsilon} e_k y_{off,i,k} + \upsilon_p^l - \mathcal{M}(1 - \sum_{k \in P^l} e_k), \quad (87)$$

$$y_{i,j} \leq y^d + \sum_{k=1,\dots\epsilon} e_k y_{off,i,k} + \upsilon_p^r + \mathcal{M}(1 - \sum_{k \in P^r} e_k), \quad (88)$$

$$y_{i,j} \geq y^d + \sum_{k=1,\dots\epsilon} e_k y_{off,i,k} + \upsilon_p^r - \mathcal{M}(1 - \sum_{k \in P^r} e_k), \quad (89)$$

where $(x_{i,j}, y_{i,j})$ is the coordinate of the microstrip endpoint that is connected to the corresponding pin, $P^z, z \in \{l,d,r,u\}$

TABLE I: RFIC layout features and spent time in design by manual design ("MAN"), the progressive ILP-based approach ("P-ILP") [14], and the proposed improved progressive ILP-based approach ("IP-ILP").

Circuit	# of microstrips	# of devices	Area(µm×µm) (ratio)	Maximum bend number			Total bend number			Runtime		
				MAN	P-ILP	IP-ILP	MAN	P-ILP	IP-ILP	MAN	P-ILP	IP-ILP
94 GHz	25	34	890×615 (1)	9	4	3	59	22	22	>2 weeks	18m05s	01m48s
LNA			845×580 (0.95)	n/a	5	4	n/a	29	30	n/a	28m13s	08m25s
			800×550 (0.9)	n/a	n/a	4	n/a	n/a	30	n/a	n/a	14m40s
60 GHz	14	26	595×850 (1)	4	3	1	27	7	6	>1 week	04m22s	00m57s
Buffer			505×720 (0.85)	n/a	3	3	n/a	13	17	n/a	19m20s	03m56s
60 GHz	19	28	600×855 (1)	4	2	2	31	10	18	>1 week	06m17s	02m31s
LNA			570×810 (0.95)	n/a	5	4	n/a	18	18	n/a	07m12s	03m46s
			540×770 (0.9)	n/a	n/a	4	n/a	n/a	24	n/a	n/a	03m40s

TABLE II: Simulation results of MAN, P-ILP, and IP-ILP.

Circuit	Area (µm×µm)	S_{11}(dB)			S_{22}(dB)			S_{21}(dB)			Bandwidth(GHz)		
		MAN	P-ILP	IP-ILP	MAN	P-ILP	IP-ILP	MAN	P-ILP	IP-ILP	MAN	P-ILP	IP-ILP
94 GHz	890×615	-27.232	-25.529	-23.537	-19.441	-21.941	-20.133	17.196	17.375	17.165	90.4–104.5	90.7–104.3	90.4–103.7
LNA	845×580	n/a	-19.517	-34.192	n/a	-14.692	-22.069	n/a	16.463	17.17	n/a	92–105	90.4–104.2
	800×550	n/a	n/a	-22.407	n/a	n/a	-21.004	n/a	n/a	16.655	n/a	n/a	89.9–104.1
60 GHz	595×850	-14.153	-14.576	-14.709	-14.342	-17.425	-15.323	16.791	17.332	17.061	45.62–68.5	47–67.3	46–68.1
Buffer	505×720	n/a	-14.444	-13.444	n/a	-17.27	-17.612	n/a	17.187	17.079	n/a	47–67.5	47.1–67.5
60 GHz	600×855	-17.511	-16.739	-16.883	-19.017	-17.297	-17.421	20.327	19.996	20.149	56–68	55.7–72.7	56–73
LNA	570×810	n/a	-16.931	-15.802	n/a	-17.319	-17.947	n/a	20.149	20.177	n/a	55.5–72.5	55.5–71
	540×770	n/a	n/a	-15.468	n/a	n/a	-18.026	n/a	n/a	20.094	n/a	n/a	55.5–75.5

is a set that each e_k belonging to it represents a combination, and the meaning of other variables and constants are the same as in constraint (21) and (22).

An example is shown in Figure 10(d), where $seg_{i,1}$ competes for area with device b. In order to avoid the overlapping of bounding boxes, the pin that $seg_{i,1}$ is connected to deviates from its target position by v_a^r.

5) Optimization formulation

The relaxation of design rules is realized by adding costs to optimization objectives. In addition to the relaxations mentioned in Section V-B, we also inherit the mircostrip length relaxation constraints from the global layout generation model. Thus, our layout validation model can be formulated as:

Minimize:

$$\alpha n_{b,max} + \beta \sum_{i \in I} n_{b,i} + \gamma \max_{i \in I} l_{rel,i} + \zeta \sum_{i \in I} l_{abs,i} + \vartheta \sum_{p \in D, k \in R} v_p^k$$
$$+ \eta \sum_{i \in I, j \in J, k \in R'} \varphi_{i,j}^k + \omega \sum_{i \in I, j \in J} \varrho_{i,j} + \delta \sum_{i \in I, j \in J} \chi_{i,j} \qquad (90)$$

Subject to: (6)–$(17), (23)$–$(28), (34)$–$(39),$

$$(43)\text{–}(51), (61)\text{–}(71), (74)-(89), \qquad (91)$$

where D is the set containing all devices, $I=\{1,\cdots,m\}$, $J=\{1,\cdots,n_i-1\}$, $R=\{l,r,d,u\}$, and $R'=\{l',r',d',u'\}$.

At the end of each iteration, we add additional chain points to microstrips where violations of design rules occur. As mentioned above, we also update the upper bound of relaxation variables in each iteration, and thus the relaxation degrees will gradually converge. The optimization process stops when all relaxation variables are minimized to zero, which means that a final layout fulfilling all design rules is generated,

VI. Experimental Results

The proposed framework was implemented with the C++ programming language, and executed on a computer with a 2.67 GHz CPU and 12 GB memory. We employed Gurobi Optimizer [15] as our ILP solver. We compare the resulting layouts generated by applying the proposed method ("IP-ILP") with the manually-designed layouts given by RFIC designers ("MAN"), and with the layouts generated by applying our prelinminary method ("P-ILP") [14]. Table I shows the layout features by applying different methods.

In Table I, for each circuit, we apply different constraints for chip size by adjusting L_h and L_v in (39) while keeping the same area ratio as the manual designs. As shown in Table I, the layouts generated by applying P-ILP and IP-ILP both remarkably surpass the manual designs for maximum number of bends, total number of bends, and runtime (for MAN, this is the time spent in manual design; for P-ILP and IP-ILP, this is the program runtime).

Compared with the results generated by P-ILP, the results generated by IP-ILP show reduction in the maximum number of bends. The maximum number of bends is an important factor that influences the circuit performance. If many bends occur on the same microstrip, the resulting serious discontinuity effect may lead to remarkable power loss, and thus drag the performance of the whole circuit down significantly.

This claim can be supported by the Agilent Advanced Design System (ADS) simulation results as shown in Table II, in which the parameter S_{11} reprensents the return loss at the input port, S_{22} reprensents the return loss at the output port, and S_{21} reprensents the voltage gain, which is the main indication for circuit performance. In Table II, both P-ILP and IP-ILP can generate layouts within similar performance as manual designs, and the results of IP-ILP are more stable and generally better than the results of P-ILP. Note that though both P-ILP and IP-ILP can generate layouts within much fewer bends than manual designs, their performance are similar, since though the length compensation caused by smoothing a bend is about 5µm and is set to 5µm in P-ILP and IP-ILP, the accurate length compensation is different for each bend according to current device placement and routing patterns. Therefore, after a long period of tuning process, a manual

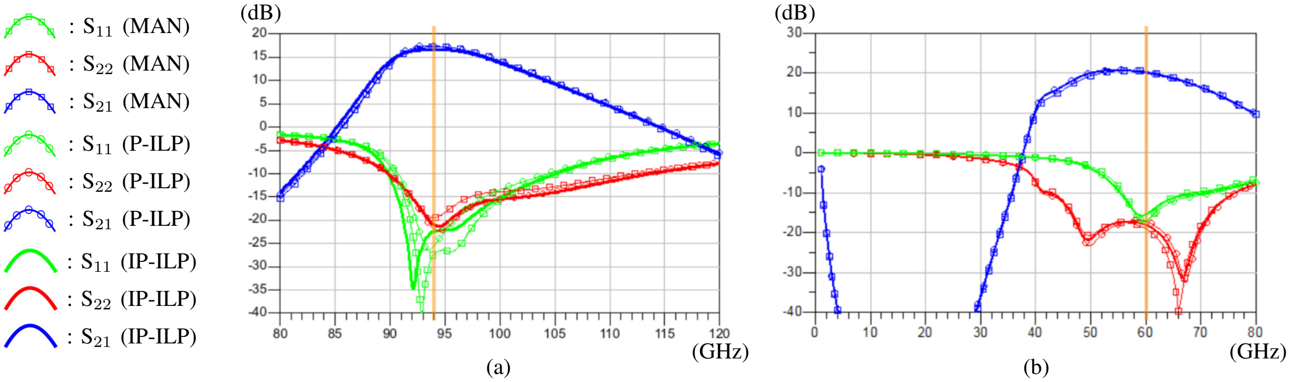
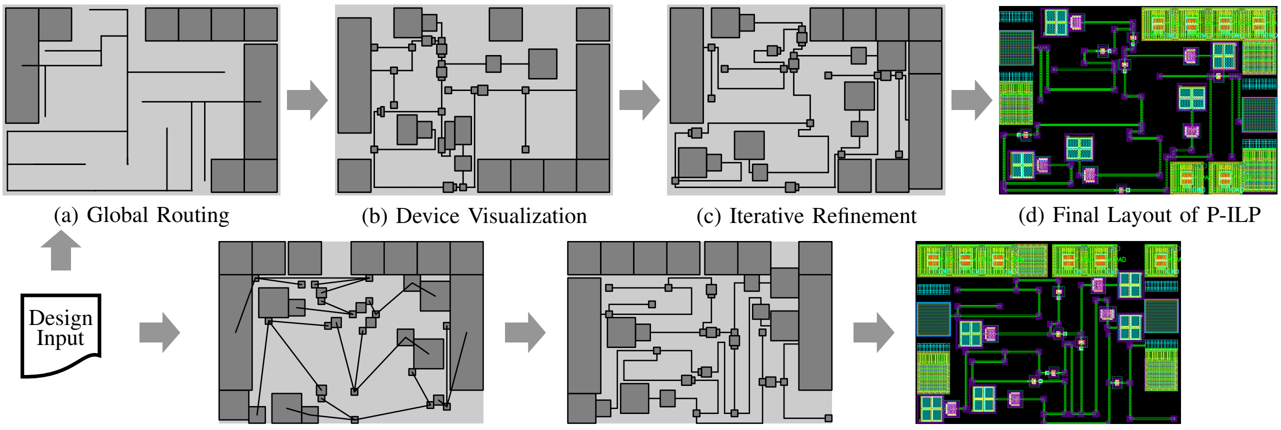


Figure 11: RF simulation results. (a) 94 GHz LNA: manual design with area 890µm×615µm, P-ILP with area 845µm×580µm, and IP-ILP with area 800µm×550µm. (b) 60 GHz LNA: manual design with area 600µm×855µm, P-ILP with area 570µm×810µm, and IP-ILP with area 540µm×770µm.



(a) Global Routing (b) Device Visualization (c) Iterative Refinement (d) Final Layout of P-ILP

(e) Global Layout Generation (f) Iterative Validation (g) Final Layout of IP-ILP

Figure 12: Snapshots of each phase for the circuit 94 GHz LNA by applying different methods. (a)(b)(c)(d): P-ILP (845µm×580µm); (e)(f)(g): IP-ILP (800µm×550µm).

design can still reach high performance if the effects brought by each bend are considered appropriately.

Figure 11 shows the simulation results under different operating frequencies. S_{11} and S_{22} seem not very stable when applying different methods under their typical working frequencies (94 GHz in Figure 11(a) and 60 GHz in Figure 11(b)), but the deviation is indeed very small, since the values of S_{11} and S_{22} are only about -15 to -20 dB, while the values of S_{21} are about 15 to 20 dB. If we focus on the plot of S_{21}, we can find that the change of S_{21} according to the change of the operating frequency for any method is consistent with that for other methods.

Runtime is the most impressive difference among MAN, P-ILP, and IP-ILP: P-ILP costs much less time than MAN, and IP-ILP costs much less time than P-ILP. Thanks to the powerful global layout generation phase in IP-ILP, the global layout is very similar to the final layout, so that the final layout can be easily obtained after a quick converge during iterative validation.

An example demonstrating the performance of our new global layout generation model is shown in Figure 12. Figure 12(a)-Figure 12(c) show the layout variation of the results generated by our previous P-ILP mehtod, and Figure 12(e) and Figure 12(f) show the layout variation of the results generated by our new IP-ILP method. Compared with the three-phase layout generation by P-ILP, our new method can efficiently generate a much more precise global layout, which only needs minor modification to be transformed into a final layout that meets all design rules. Therefore, the loading for the later iterative layout validation phase is remarkably alleviated, and thus the program run time can be significantly shortened.

Another benefit brought by the global layout generation is an even distribution of devices and microstrips. In the global routing phase of P-ILP, devices except pads are blurred and thus the area reservation for devices may be over- or under-estimated. However, in the global layout generation phase of IP-ILP, though devices are simplified as squares, the deviations of device dimensions are minor as shown in Figure 12(e) and Figure 12(f). With a better resource allocation supported by the global layout generation model, devices and microstrips are distributed evenly on the chip as shown in Figure 12(f).

VII. Conclusion

In this paper, we propose an improved progressive integer-linear-programming (IP-ILP) method to generate high-quality layout for mm-wave RFICs. In RFIC circuit routing, microstrip lines need to match the given lengths to maintain circuit performance. In addition, bends on microstrips should be reduced as much as possible. We model this layout generation task as an ILP problem and solve it in two phases: global layout generation and iterative validation. Experiments show that the proposed method can generate a valid layout efficiently, while circuit performance resulting from the automatically-generated layouts is consistent with the performance of manual designs, even within smaller chip area. Compared with our preliminary work, our new method shows a remarkable reduction of program runtime. Moreover, we further reduce the maximum number of bends, and we distribute devices and microstrips more evenly in the final layout.

Acknowledgment

The authors would like to thank National Chip Implementation Center (CIC) for the software and supporting.

References

[1] C.-W. Byeon, C.-H. Yoon, and C.-S. Park, "A 67-mW 10.7-Gb/s 60-GHz OOK CMOS transceiver for short-range wireless communications," *IEEE Trans. Microw. Theory Tech.*, vol. 61, no. 9, pp. 3391–3401, 2013.

[2] D. Fritsche, G. Tretter, C. Carta, and F. Ellinger, "Millimeter-wave low-noise amplifier design in 28-nm low-power digital CMOS," *IEEE Trans. Microw. Theory Tech.*, vol. 63, no. 6, pp. 1910–1922, 2015.

[3] F. Schnieder and W. Heinrich, "Model of thin-film microstrip line for circuit design," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 1, pp. 104–110, 2001.

[4] C.-M. Lo, C.-S. Lin, and H. Wang, "A miniature V-band three-stage cascode low noise amplifier in 130-nm CMOS technology," in *Int. Solid-State Circuit Conf.*, 2006, pp. 1254–1263.

[5] D. M. Pozar, *Microwave Filters*. Wiley, 1998.

[6] M. Aktuna, R. A. Rutenbar, and L. R. Carley, "Device-level early floorplanning algorithms for RF circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 4, pp. 375–388, 1999.

[7] M. Pathak and S. K. Lim, "Fast layout generation of RF embedded passive circuits using mathematical programming," *IEEE CPMT*, vol. 2, no. 1, pp. 32–45, 2012.

[8] Y. Kubo, H. Miyashita, Y. Kajitani, and K. Takeishi, "Equidistance routing in high-speed VLSI layout design," *Integr. VLSI J.*, vol. 38, no. 3, pp. 439–449, 2005.

[9] T. Yan and M. D. F. Wong, "BSG-Route: A length-matching router for general topology," in *Proc. Int. Conf. Comput.-Aided Des.*, 2008, pp. 499–505.

[10] M. M. Ozdal and R.-F. Hentschke, "Exact route matching algorithms for analog and mixed signal integrated circuits," in *Proc. Int. Conf. Comput.-Aided Des.*, 2009, pp. 231–238.

[11] K.-H. Ho, H.-C. Ou, Y.-W. Chang, and H.-F. Tsao, "Coupling-aware length-ratio-matching routing for capacitor arrays in analog integrated circuits,," in *Proc. Design Autom. Conf.*, 2013, pp. 1–6.

[12] T.-M. Tseng, B. Li, T.-Y. Ho, and U. Schlichtmann, "Post-route alleviation of dense meander segments in high-performance printed circuit boards," in *Proc. Int. Conf. Comput.-Aided Des.*, 2013, pp. 713–720.

[13] ——, "Ilp-based alleviation of dense meander segments with prioritized shifting and progressive fixing in pcb routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 6, pp. 1000–1013, 2015.

[14] T.-M. Tseng, B. Li, C.-F. Yeh, H.-C. Jhan, Z.-M. Tsai, M. P.-H. Lin, and U. Schlichtmann, "Novel cmos rfic layout generation with concurrent device placement and fixed-length microstrip routing," in *Proc. Design Autom. Conf.*, 2016, pp. 101:1–101:6.

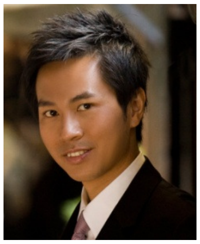
[15] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com

Tsun-Ming Tseng received the bachelor degree in electronics engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2010, and the master degree in communications engineering from Technical University of Munich (TUM), Munich, Germany, in 2013. He is currently pursuing a Dr.-Ing. degree at the Institute for Electronic Design Automation, TUM. His research interests focus on large-scale design automation.

Bing Li received the bachelors and masters degrees in communication and information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2000 and 2003, respectively, and the Dr.-Ing. degree in electrical engineering from Technical University of Munich (TUM), Munich, Germany, in 2010. He is currently a researcher with the Institute for Electronic Design Automation, TUM. His research interests include timing and power analysis as well as emerging systems.

Ching-Feng Yeh received the M.S. degree in the Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan in 2016. His research interest lies in analog/RF layout automation.

Hsiang-Chieh Jhan was born in Chiayi, Taiwan, in 1993. He received a B.S. degree 2015 from the Department of Electrical Engineering in National Chung Cheng University. He is currently working toward the M.S. degree at National Chung Cheng University, Chiayi, Taiwan. His research interests include the design of microwave integrated circuit, microwave systems.



Zuo-Min Tsai was born in Miaoli, Taiwan, in 1979. He received a B.S. degree 2001 from the Department of Electrical Engineering in National Taiwan University and a Ph.D. degree in communication engineering from National Taiwan University, Taipei, Taiwan, in 2006. From 2006-2011, Dr. Tsai was postdoctoral research fellow with the Graduate Institute of Communication Engineering, National Taiwan University. In July 2011, he joined the faculty of the Department of Electrical Engineering, National Chung Cheng University, where he is currently an associate professor. His research interests include the design of microwave integrated circuits and microwave systems.

Mark Po-Hung Lin (S'07–M'09–SM'13) received the B.S. and M.S. degrees in electronics engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, respectively, and the Ph.D. degree in the Graduate Institute of Electronics Engineering, National Taiwan University (NTU), Taipei, Taiwan. He has been with the Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan, since 2009, where he is currently an Associate Professor. He was with SpringSoft, Inc. (acquired by Synopsys, Inc. in 2012) during 2000–2007 as an Engineer, a Senior Engineer, an Associate Manager, and a Technical Manager. He was a Visiting Scholar with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, during 2007–2009, a Humboldt Research Fellow with the Technical University of Munich (TUM), Germany, during 2013–2015, and a JSPS Invitation Fellow with Osaka University, Japan, in 2016.

Dr. Lin's research interests include design automation for analog/mixed-signal integrated circuits and low-power resilient circuit and system design optimization. He served in the technical program committees of many premier conferences, including ACM/IEEE Design Automation Conference (DAC), ACM/IEEE Design, Automation & Test in Europe (DATE), ACM/IEEE Asia South Pacific Design Automation Conference (ASP-DAC), and ACM International Symposium on Physical Design (ISPD).

Dr. Lin was recipients of IEEE Tainan Section Macronix Award, IEEE Tainan Section Best GOLD Member Award, Humboldt Research Fellowship for Experienced Researchers, JSPS Invitation Fellowship for Research in Japan, Distinguished Young Scholar Award of Taiwan IC Design Society, Outstanding Young Electrical Engineer Award of the Chinese Institute of Electrical Engineering, and Distinguished Young Faculty Award of National Chung Cheng University.

Ulf Schlichtmann (S'88–M'90) received the Dipl.Ing. and Dr.Ing. degrees in electrical engineering and information technology from Technical University of Munich (TUM), Munich, Germany, in 1990 and 1995, respectively. He was with Siemens AG, Munich, and Infineon Technologies AG, Munich, from 1994 to 2003, where he held various technical and management positions in design automation, design libraries, IP reuse, and product development. He has been with TUM as a Professor and the Head of the Institute for Electronic Design Automation, since 2003. He served as the Dean of the Department of Electrical Engineering and Information Technology, TUM, from 2008 to 2011. His current research interests include computer-aided design of electronic circuits and systems, with an emphasis on designing reliable and robust systems.