Analog Layout Design Automation: ILP-Based Analog Routers

by
©Mohammad Torabi
A dissertation submitted to the School of Graduate Studies
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Faculty of Engineering & Applied Science
Memorial University of Newfoundland

Supervisory Committee
Dr. Lihong Zhang (Supervisor)
Dr. Tariq Iqbal
Dr. Ramachandran Venkatesan

October 2018

St. John’s, Newfoundland

Abstract

The shrinking design window and high parasitic sensitivity in the advanced technology have imposed special challenges on the analog and radio frequency (RF) integrated circuit design. In this thesis, we propose a new methodology to address such a deficiency based on integer linear programming (ILP) but without compromising the capability of handling any special constraints for the analog routing problems. Distinct from the conventional methods, our algorithm utilizes adaptive resolutions for various routing regions. For a more congested region, a routing grid with higher resolution is employed, whereas a lower-resolution grid is adopted to a less crowded routing region. Moreover, we strengthen its speciality in handling interconnect width control so as to route the electrical nets based on analog constraints while considering proper interconnect width to address the acute interconnect parasitics, mismatch minimization, and electromigration effects simultaneously. In addition, to tackle the performance degradation due to layout dependent effects (LDEs) and take advantage of optical proximity correction (OPC) for resolution enhancement of subwavelength lithography, in this thesis we have also proposed an innovative LDE-aware analog layout migration scheme, which is equipped with our special routing methodology. The LDE constraints are first identified with aid of a special sensitivity analysis and then satisfied during the layout migration process. Afterwards the electrical nets are routed by an extended OPC-inclusive ILP-based analog router to improve the final layout image fidelity while the routability and analog constraints are respected in the meantime. The experimental results demonstrate the effectiveness and efficiency of our proposed methods in terms of both circuit performance and image quality compared to the previous works.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Lihong Zhang for his continuous support in my PhD study, research, motivation, enthusiasm, immense knowledge and his assistance in writing papers and thesis. I also intend to thank the other members of my supervisor committee, Dr. Venkatesan and Dr. Iqbal, for their guidance and suggestions.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, in part by Canada Foundation for Innovation, in part by the Research and Development Corporation of Newfoundland and Labrador through its Industrial Research and Innovation Fund and ArcticTECH R&D Award, and in part by the Memorial University of Newfoundland.

To my lovely wife

Table of Contents

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Analog Constraint Specific	**ACS**
Electronic Design Automation	**EDA**
Dual Stress Liner	**DSL**
Electromigration	**EM**
Integer-Linear-Programming	**ILP**
Integrated Circuits	**IC**
Layout Dependent Effect	**LDE**
Length of Oxide Diffusion	**LOD**
Low-Noise Amplifier	**LNA**
Optical Proximity Correction	**OPC**
Poly Space Effect	**PSE**
Radio Frequency	**RF**
Resolution Enhancement Technique	**RET**
Two-stage Operational Amplifier	**OPAMP**
Well Proximity Effect	**WPE**
White Space	**WS**

Chapter 1 Introduction

The state-of-the-art VLSI technologies feature highly complex and extremely expensive manufacturing processes. To reduce errors in the design stage, enhance performance, and boost productivity, electronic design automation (EDA) tools have to be extensively used. As an indispensable part of many modern system-on-chip designs, analog and RF circuits, which are becoming increasingly more complex due to high performance demand, have been getting more intractable and error prone in the time-consuming iterative design process. This is because they are normally more sensitive to parasitics and manufacturability issues compared to their digital counterparts. To address these issues common in the traditional manual design style, EDA tools are called upon to play a more important role in the design of analog and RF circuits to offer better performance and sound productivity.

A widely accepted analog/RF circuit design flow is illustrated in Fig. 1 [1]. As shown in this figure, after defining the circuit specifications, the first stage is called "*Topology Synthesis*", which is aimed to find the best arrangement of the circuit devices that can contribute to the best circuit performance. The output of this stage is a schematic or netlist that contains the connectivity information of the circuit devices. Afterwards, the circuit device sizes and biasing conditions are optimized in the "*Sizing*" stage for the objectives of the best circuit performance and the least total layout area. Both the topology synthesis and sizing operation are normally categorized into the circuit-level design, while the other stages (i.e., "*Physical Layout Design*", "*Parasitic Extraction*", and "*Verification*") fall in the layout-level design [1].

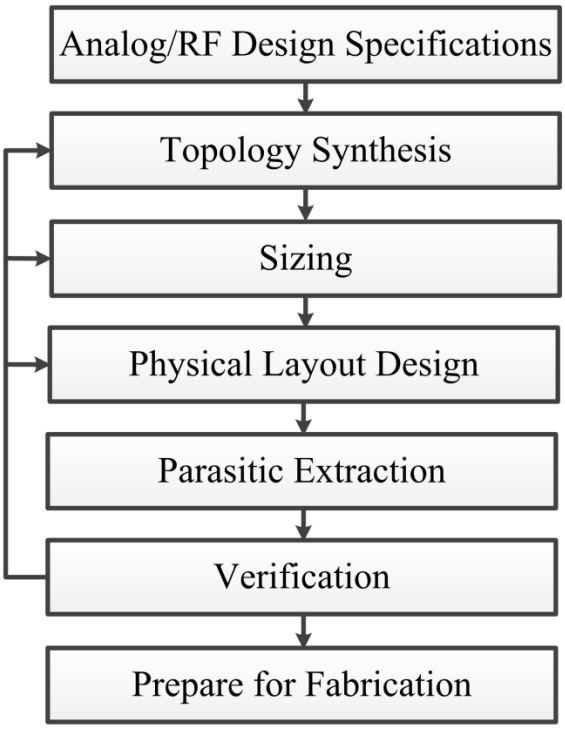


Fig. 1. Analog/RF circuit design flow [1].

Physical layout design, also called layout synthesis, is typically comprised of module generation, placement, and routing. All the devices or functional building blocks in an analog or RF circuit are first converted to their corresponding module layouts. This process is called module generation [2] [3], which is to produce primitive units for the consequent operations. Digital circuits normally use standard cell libraries for their modular units, while analog/RF circuits need more sophisticated modules to be customized for their special functionality in the context of various analog constraints. For instance, common centroid structure is a technique in the analog layout design for mitigating the effect of process gradient. Thus, it is necessary for the module generator to consider the common centroid structures. Reference [3] characterized different types of devices (such as transistors, capacitors, and resistors) along with the user-defined specifications to generate the modules. This work considers mismatch and capacitance in the module generation step to minimize their effects on the circuit performance. In [4], the

proposed module generator starts from several pre-generated cells. If a matching constraint is required, suitable modifications would be applied to the cells.

The placement process is to locate all the primitive modules to proper spots under demanded constraints. The major placement constraints are layout area and analog constraints, although some works consider wirelength as another constraint to decrease the interconnect parasitic impact on circuit performance. The main analog placement constraints are symmetry, common centroid, proximity, regularity, etc. [5]. Symmetry, as one of the most important analog matching constraints, aims to ensure that symmetric devices are placed symmetrically with reference to the symmetry axis. For instance, differential pair and current mirror are two typical building blocks in the analog circuits that comprise the feature of symmetry. This constraint has been addressed by several previous works [6] [7] [8] [9] [10] [11]. As for the common centroid constraints [6] [12] [13], devices are first decomposed into multiple units that would then be evenly arranged into a two-dimensional array with respect to a common centroid point.

Moreover, the proximity constraints [6] [7] are to restrict the distances among devices, while the regularity constraints [14] [15] are to arrange the devices into rows or arrays so as to make the final layout less sensitive to process variation but better routability. In the placement algorithms, data structure used for representing the device topology can strongly affect the algorithmic efficiency and flexibility of handling the analog/RF constraints. In this regard, B* tree, sequence pair, slicing tree, and transitive closure graph are common methodologies for representing the relative position of devices. A survey on different data structures used by analog placement in recent years is presented by [5]. The core of such placement algorithms is to perturb different possible solutions according to the representation structure with respect to the analog constraints.

The analog routing process is to connect electrical terminals of the primitive modules by using optimal paths to meet the routing constraints for reaching the best performance. As the final step in the layout synthesis [1], routing normally encloses two internal stages for the analog and RF layouts. As the first stage, global routing plans interconnection of the terminals based on the routing grids. The second stage is called detailed routing, which is aimed to physically realize the interconnection of the netlist precisely based on the solution proposed by the global routing.

The performance of the analog and RF circuits may be strongly affected by the routing process [16]. For instance, a proper routing operation may smartly balance the routing congestion and meanwhile reduce distance between certain sensitive nets or devices, which can effectively control the related parasitic effects that may otherwise ruin the circuit performance [17]. Moreover, the routing quality is significant for consolidating the advantage or making up the deficiency from the previous placement stage with respect to some special constraints, such as symmetry and common centroid [18], in the entire layout synthesis. Therefore, it is critical for an analog router to be able to handle these analog constraints in order to secure the performance of analog and RF integrated circuits. In this thesis, to deal with the analog/RF constraints, we have proposed routing methodologies to address analog/RF issues such as routing efficiency, electromigration, interconnect parasitics, mismatch minimization, layout dependent effects (LDE), and optical proximity correction (OPC) along with other analog and routability constraints.

As shown in Fig. 1, the second last stage of analog/RF circuit design flow is parasitic extraction that derives the parasitics of the layout for both interconnects and devices. Since analog circuits suffer from parasitic effects, it is essential to verify the functionality of the final layout after parasitic extraction. Thus, this verification process checks whether the circuit

specifications are satisfied in the presence of parasitics. If so, the layout is ready for fabrication. Otherwise, certain iterations in the design flow are needed to address the drawbacks within the design.

The rest of the thesis is organized as follows. Chapter 2 reviews analog/RF EDA challenges, the definition of analog routing problem, and the previous related works in the analog/RF domain. Chapter 3 describes the adaptive routing resolution methodology for efficiency enhancement in the proposed router. In Chapter 4, EM- and interconnect-parasitic-aware routing method and mismatch minimization are explained. An LDE-aware analog retargeting and OPC-inclusive routing method are discussed in Chapter 5. A conclusion is drawn in Chapter 6, which is followed by a discussion about future work in Chapter 7.

Chapter 2 Analog Design Automation, Challenges and Solutions

In this chapter, the recent challenges of analog design automation, which include interconnect parasitics, electromigration, LDE, and OPC, will be discussed in Section 2.1. Afterwards in Section 2.2, we will provide a literature survey of the solutions to addressing the analog design automation challenges.

2.1. Challenges in Analog Design Automation

Since the last century, electronic design automation (EDA) tools have been actively developed to help reduce circuit designers' errors and improve circuit performance. As modern integrated circuits (ICs) are becoming increasingly more complex and larger, powerful EDA tools are required to handle complicated design rules and circuit constraints for providing optimal results efficiently. Although many state-of-the-art EDA tools have been quite successful in the domain of digital circuits, more efforts are still needed for analog and RF circuits because of their high performance-sensitivity in addition to the other effects as well as inherent error-prone analog/RF design process. In this section, the most recent and important analog/RF EDA challenges have been identified and discussed.

2.1.1. Electromigration (EM)

Interconnect and via failure due to the EM effect has been a major issue as a result of shrinking feature size in the advanced technologies. EM causes open circuit in metal

interconnects as well as void in metal-to-metal vias [19]. In the literature, the *mean-time-to-failure (MTTF)*, which is described by Black's Equation [20], is used as a criterion metric for EM to represent the mean lifetime as shown in (1),

$$MTTF = \frac{A}{j^n} \exp(\frac{E_a}{KT}), \quad (1)$$

where A is a technology material constant, j is the current density, exponent n is a value between 1 and 2, E_a is the activation energy, K is the Boltzmann's constant, and T is the film temperature. Besides the current density, wirelength is also a parameter that should be considered for EM. The wirelength for one interconnect segment immune from the EM issues can be described by Blech length effect, which expresses that the interconnect segment with certain wirelength shorter than the Blech length is called EM-immortal. The Blech length equation [21] can be described as follows:

$$(JL)_{Critical} = \frac{\sigma_{\max}\Omega}{Z^* e\rho}, \quad (2)$$

where J is the current density, L is the wirelength, σ_{max} is the maximum back-stress, Ω is the atomic volume, Z^* is the effective valence, e is the electron's charge, and ρ is the resistivity. In (2) above, product JL is the critical JL (i.e., $(JL)_{Critical}$) when the maximum back-stress is generated for the interconnect segment. $(JL)_{Critical}$ is assumed to be a constant for one specific technology. Thus, the Blech length effect can be stated in other words, that is, "one interconnect segment with current density j is EM-immortal when its wirelength l is less than $\frac{(JL)_{Critical}}{j}$" or "one interconnect segment with product jL less than $(JL)_{Critical}$ is EM-immortal".

2.1.2. Layout Dependent Effect (LDE)

Due to the huge impact on analog circuit performance, LDE should be seriously addressed for analog circuits in the advanced CMOS technologies. This is mainly because some electrical characteristics of the MOSFET devices such as threshold voltage and mobility vary a lot due to the change in the LDE parameters. Thus, the transistor current may change accordingly to cause the circuit performance degradation if the LDEs are not seriously considered in the analog layout design [22]. The most dominant LDE sources are well proximity effect (WPE) and shallow trench isolation (STI) [23] that will be discussed in the following subsections.

2.1.3. Well Proximity Effect (WPE)

When a transistor is located near the edge of the photoresist, during the manufacturing the dose in the nearby transistors could increase because the dopant ions might scatter out of the resist. This can vary the threshold voltage and other transistor characteristics depending on the location and shape of the adjacent wells [24] and this effect is known as well proximity effect. Thus, MOSFET channel may be graded if it is placed close to the well edge as a result of WPE.

WPE can be expressed by the following BSIM model [25].

$$Vth0 = Vth0_{org} + KVTH0WE * (SCA + WEB * SCB + WEC * SCC), \quad (3)$$

$$K2 = K2_{org} + K2WE * (SCA + WEB * SCB + WEC * SCC), \quad (4)$$

$$\mu_{eff} = \mu_{eff_{org}} * (1 + KU0WE * (SCA + WEB * SCB + WEC * SCC)), \quad (5)$$

where *SCA*, *SCB*, and *SCC* are the instance parameters, *KVTH0WE*, *K2WE*, and *KU0WE* are the technology dependent threshold shift factor, *K2* shift factor, and mobility degradation factor for

WPE, respectively. *WEB* and *WEC* are just the coefficients for *SCB* and *SCC* [25]. A review on the BSIM model regarding WPE as well as the related works has been reported in [23].

2.1.4. Shallow Trench Isolation (STI)

The shallow trench is formed during the process of transistor isolation by etching into the wafer and filling with silicon oxide as isolation between active areas. This exerts mechanical force, which is a compressive stress applying to the vicinities, i.e., diffusion areas [23]. The stress-related parameters that describe STI are SA, SB and width of STI (i.e., STIW) as shown in Fig. 2. SA (or SB) is the distance between the Active edges and each polysilicon finger. The STIW is the oxide-to-oxide area spacing. Some literature uses *Length of Diffusion* (*LOD*) term instead of SA (SB), which reflects the stress due to the Active area length. The stress can be expressed by some equations according to the BSIM model [25]. It is clear that the smaller SA (SB) the higher stress would be applied to the transistor. The stress results in lower (higher) current in NMOS (PMOS) transistors [22] that may cause performance degradation of the analog/RF circuits.

$$Stress = 1/(SA + L/2) + 1/(SB + L/2) \ , \tag{6}$$

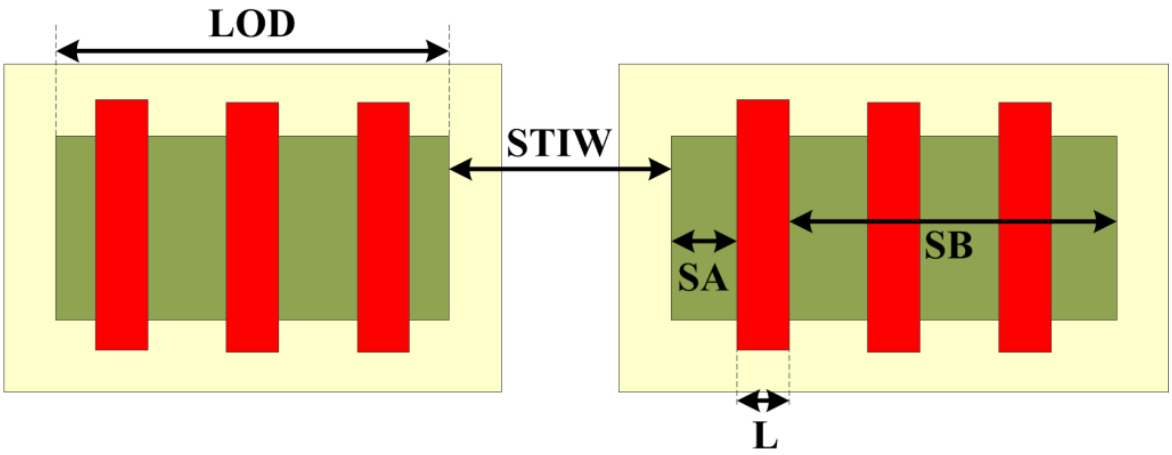


Fig. 2. Stress-related parameters of STI.

2.1.5. Optical Proximity Correction (OPC)

Modern IC technologies are actually manufacturing devices with smaller sizes than the lithography wavelength. The sub-wavelength lithography cases a huge burden to the manufacturing process since the diffraction of light physically limits the critical dimension such as the shortest length of the gate channel. The distortion in optical lithography includes corner routing, variation in line width, and narrow end of line [26]. OPC, Phase Shift Masking (PSM) and multiple patterning have been introduced as the major remedies for such layout distortions to making 193nm lithography as standard even for technologies below 65nm [27].

As one of the most popular RET techniques in the sub-wavelength era, OPC is capable of generating high layout image fidelity by modifying the polygons in the layout so as to minimize the difference between the ideal layout and the wafer image. There are two main OPC methods, rule-based and model-based. The rules-based OPC uses pre-generated patterns to produce the target layout according to a set of rules. In contrast, as for the model-based OPC methods, the circuit area is divided into several parts and the target layout is generated based on the lithography model. The advantage of the model-based OPC is higher image fidelity compared to

the rule-based one. However, the model-based OPC suffers from low efficiency due to its iterative nature in addition to high mask complexity. An example of Fig. 3 [28] illustrates how OPC can improve the image quality by manipulating the mask layout.

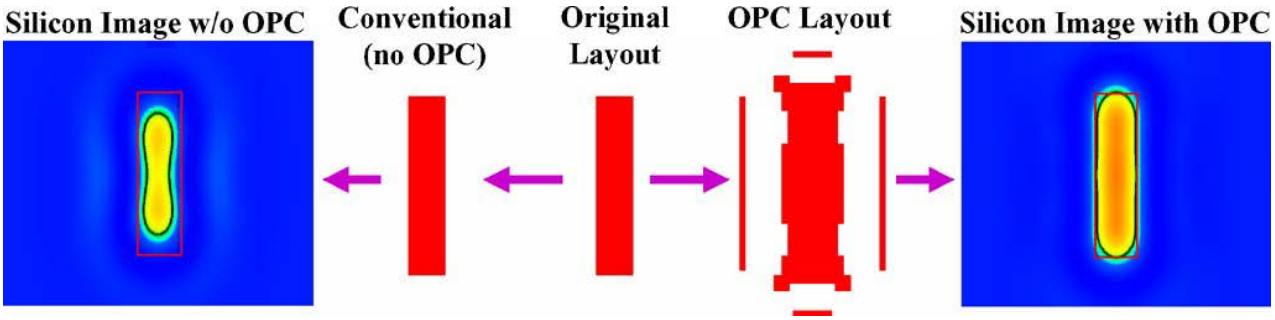


Fig. 3. An example of OPC that shows image quality improvement due to OPC [28].

2.2. State-of-the-Art Analog Routers

In this section, the definition of analog routing problem is firstly discussed in Section 2.2.1. Then the state-of-the-art routers will be reviewed and their solutions will be discussed in Section 2.2.2. Since the digital and analog routers have similar challenges in certain areas, here we will try to include several digital solutions to complete the literature review.

2.2.1. Analog Routing Problem Definition

The analog routing problem definition will be explained briefly below and later in Chapter 3 the formulation will be described in more detail. Suppose that $G=(V,E)$ is a three-dimensional graph, in which V is the set of vertices and E is the set of edges, and set $N=\{0,..,n-1\}$ includes a list of the electrical net indices. Two kinds of vertices exist in the graph: terminals and non-terminals. The *terminals* include circuit input/output pins and device pins. Each electrical net has a list of terminals that need to be connected with one another. That is to say, all the vertices in

the graph belonging to the netlist are terminals. A netlist is a description of the connectivity of an electronic circuit. The *non-terminals* are the intermediate vertices in the graph, which are used to connect the terminals. According to the description above the input to the analog routing problem is a netlist and its output is the paths in graph *G* among all terminals in the netlist. Thus, the analog routing problem can be formally described as "*finding the most suitable set of edges in graph G that can define paths among the terminals of the electrical nets besides satisfying the routability and analog constraints*".

For example, a routing graph is shown in Fig. 4 (a), where *Net1* (*Net2*) has to be routed between source *S1* (*S2*) and sink *T1* (*T2*). In this example, *S1* (*S2*) and *T1* (*T2*) are terminals and the other vertices are non-terminals. If *Net1* and *Net2* are required to be symmetric, the legal solution shown in Fig. 4 (a) can satisfy the routability and symmetry constraints. Fig. 4 (b) shows eight segments emanating from each vertex in the graph.

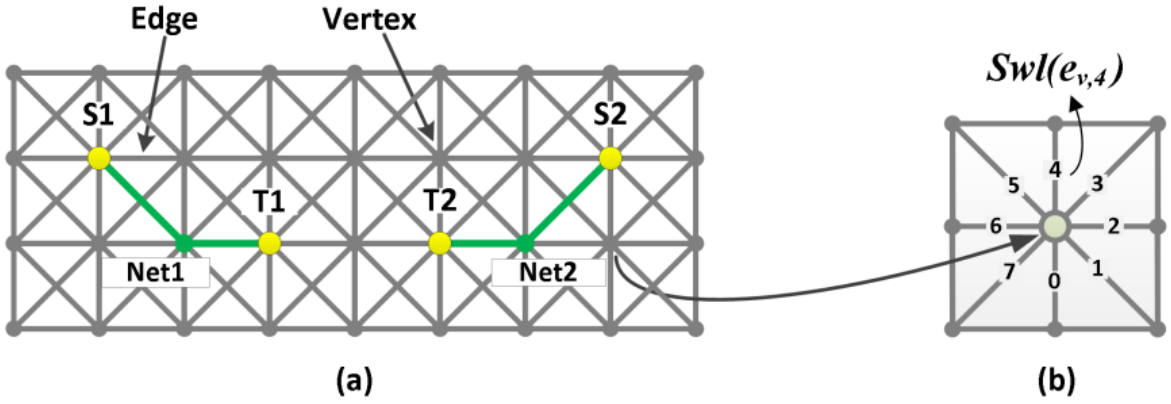


Fig. 4. (a) Routing solution for *Net1* and *Net2*. (b) Segment indices for one vertex *v* and segment wirelength $Swl(e_{v,s})$ when s=4.

2.2.2. Recent Analog Routers

In the literature, two groups of analog routers have been developed for distinct purposes. The first group tries to offer a general-purpose analog router, which is able to route all the nets in a circuit while satisfying general analog constraints. So a special focus has to be put onto the coordination of multiple nets routing and the balanced consideration between routability and efficiency. Based on the routing sequence of various nets, this category can be further split into: 1) sequential analog routers that route different nets one after another based on a priority list; 2) concurrent analog routers that route all the nets simultaneous. In contrast, the second category, which pays less attention to routing the entire netlist in a circuit, is concentrated on satisfying specific or extensive analog constraints, such as exact wirelength, mismatch effect, electromigration, and IR-drop. We call this category as analog constraint specific (ACS) routers, which are supposed to be integrated into another general-purpose router for practical usage towards the routing duty within analog layout synthesis.

The history of the general-purpose analog routers can be traced back to the early dates of analog EDA. A milestone sequential analog routing tool ANAGRAM II was integrated into a complete academic layout synthesis suite. Utilizing a wire expanding algorithm [29], this analog router can search for a path with the lowest cost in terms of wirelength, crosstalk, as well as other user-defined analog constraints. Along with its subsequent updates, this work eventually turned to one of the earliest commercial analog layout tools. Within the latest decade, Yilmaz and Dundar developed an analog layout generator enclosing a two-step routing scheme highly applicable to analog CMOS circuits [3]. Although the proposed formulation does include circuit parasitics, the sequential routing scheme may not guarantee that all the claimed analog constraints can be simultaneously satisfied. Reference [30] is focused on a combination of

placement and routing techniques together in the analog layout design. The routing of symmetric nets is managed by a conventional mirroring scheme, which routes the first part of a symmetric net and then follows a symmetric path for the counterpart to satisfy the symmetry constraints. Afterwards a process of layer assignment is deployed to distribute physical layer resources to various wires. Despite the prominent routing results for analog circuits, the net priority, which may affect the performance of certain sensitive analog circuits, has not been considered in their work.

By using integer linear programming (ILP), Ou et al. proposed a router that can handle several constraints, including symmetry, topology matching, and exact length matching [31]. In spite of the proposed thorough modeling for analog constraints, this work tends to experience low efficiency due to lack of special strategies to overcome the high time complexity of LP. Pan *et al.* [32] proposed a configurable analog router, which used a unified constraint method from schematic to layout. After placement, this tool can recursively partition the layout according to the constraints driven from placement and then route the nets inside each partition. However, it is not clear how wire bend and via minimization can be considered in this work.

In [33], Gao *et al.* utilized an A* algorithm and maze router to wire the interconnects, which are classified into critical and general nets. They applied net priority to the sensitive nets and further protected them by shielding the wires. Although this method intends to reduce the coupling capacitance, it may increase the number of layers required for routing, as the net classification scheme alone is hard to guarantee the optimal solution for analog layouts. In [34], a global routing methodology was proposed by first generating a set of route candidates, which are represented in a compatibility graph. Then the graph is reduced to find the best paths that can

satisfy routability and other analog constraints. This methodology would be more practical if the handling of net priority could be merged into the route candidate selection process.

Reference [35] utilizes Boolean expressions to address the digital routing problem by representing the rules with the aid of a satisfiability formulation. Afterwards the routing results can be obtained by solving the satisfiability model. Although this work exposes an interesting research means, it is not clear whether such a formulation is applicable to modeling analog routing constraints. MARS [36] is another digital router that utilizes a multilevel hierarchical scheme to overcome the huge size of routing area. Different from the conventional methods with either upward or downward routing flow, this work manages the routing in two sequential stages, downward flow first and then upward flow. In the downward flow from the finest to the coarsest, the tiles are routed, while the upward flow will then refine the routing results to determine the final solution. It would be interesting if an analog general-purpose router could follow a similar scheme to handle the dilemma of rip-up and re-route. Chu and Wong developed a lookup-table-based method called FLUTE [37] to find rectilinear Steiner minimal tree, and later an obstacle-avoid version based on FLUTE was proposed in [38] to be used as a digital global router. Although it is possible to get these schemes be utilized in a general-purpose analog router, certain trimming work is still required to account for special analog constraints.

Ozdal and Hentschke [39] first conducted an algorithmic study of ACS routers with a focus on the exact route matching constraints. They proposed a mathematical formulation, based on which dynamic programming and heuristic search algorithms were developed for analog and mixed-signal integrated circuits. This work has triggered a stream of research activities on ACS routers in the subsequent years.

A gridless router proposed by [40] used a similar mathematical model for the exact route matching problem. An A* search algorithm was deployed to expand the exactly-matched nets more precisely based on the defined cost function. However, its time and space complexity is too high in practice. To address this disadvantage, reference [41] presented improved work of a constrained-path based maze routing algorithm that could handle exact matching constraints for multiple nets. A scalable framework, which utilized the proposed maze routing algorithm for realistic problem sizes, was also developed. Although the proposed mathematical model is very effective for exact route matching, it is not straightforward to extend this model to cover other analog constraints, e.g., symmetry or coupling noise. Yao *et al.* proposed another ACS router called LEMAR [42] based on wire detouring and A* algorithm to find a wirelength matching path for two nets. However, it is only limited to single layer routing scenario. It would be interesting if this work could be extended to manage multi-layer situation and include a scheme for minimizing bend and via numbers.

As an important research topic, electromigration (EM) is getting more challenging in the context of feature size shrinking and interconnect length increment in the new technologies [43] [19]. Bigalke and Lienig [44] proposed a post-processing algorithm that can improve the layout reliability against EM failure by changing the size of the routed wires and vias. In this recent work, the proposed redundant via insertion algorithm gives more consideration priority to the interconnections with higher current density for reducing the EM effect on vias. As a matter of fact, it would be more practical for the designers if the works above could be included into a general-purpose analog router to satisfy the EM constraints in the analog/RF circuits.

Moreover, EM and IR-drop have been addressed in [45] for multiport multi-terminal analog circuits by setting different wire widths, which are determined by the amount of current flow in

each branch. This work generates obstacle-avoiding minimum paths when multi-terminal devices are utilized in the analog circuits. However, although the proposed router is generally able to find an optimal path with the minimum wirelength, no EM Blech length effect [21] was considered and no optimal solution could be found for the matching nets that have to overlap with each other due to the constraints in the terminal configuration.

Li *et al.* [46] studied the impact of different via arrangement and configuration when redundant vias are needed due to the EM constraints. In [47], a model for EM lifetime of via-arrays was provided to balance the power grid integrity, reliability, and total wire area. However, it would be more interesting if this work could be integrated into a global or detailed router. A general mathematical equation was proposed in [48], where Black's equation was refined to compensate for the lack of Blech length effect. Reference [49] proposed WiT, an EM-aware wiring topology generation scheme for global routing by modeling the problem on a multi-source multi-sink flow network. Although it can find an optimal solution for the global routing purpose, more efforts are still needed to include its formulation into a detailed router especially in the context of analog/RF applications.

Tsai *et al.* [50] introduced channel space restriction into their wiring topology for EM-aware routing. By using violation and reserved-path finding scheme, they set up an LP formulation to achieve wire area minimization. Reference [51] proposed to first divide the routing area into individual routing groups, and then route them independently by an exhaustive algorithm. In spite of such a divide-and-conquer paradigm for the EM-aware routing, the relationship between the routing quality and division resolution was not studied in that work.

Jerke *et al.* [52] proposed a post-layout modification stage, which could boost the interconnection robustness against the electromigration failure. It changed the size of wires and

vias based on current flow density computation. This work tends to be still valuable to the advanced technologies as reviewed by [43] regarding the electromigration effects on the modern physical design in the context of feature size shrinking and interconnect length increment in the new technologies. Nevertheless, it is sensible to integrate this method into a general-purpose analog router to ensure the parasitics of sensitive nets would not affect circuit performance even though the size of certain wires and vias might be enlarged in the optimized analog layouts.

In the literature, several papers have been published to address LDEs as an emerging issue in the advanced physical design. Layout-dependent stress modeling was reported in [53] and [54] to address the STI issues. Compact models of stress-induced mobility, velocity, and threshold voltage were proposed to model the scalability to layout, temperature and other device parameters for different strain technologies. However, the proximity effects from the neighboring devices were not included in these works. Drennan *et al.* investigated the significance of WPE and STI in the context of analog design [55]. This milestone work has actually triggered the LDE research in the analog EDA in the recent decade.

Youssef *et al.* [56] proposed a layout generation scheme to compute and compare parasitic and STI effects for different device layout styles. A Python-based tool was developed to conduct geometric and electrical optimization by linking a sizing phase and a layout generation phase together. Nevertheless, this proposed method is only aimed for basic analog building blocks (e.g., differential pairs) rather than more complex analog circuits. A finite-element-method-based stress simulator was developed by [22] to extract the stress distribution in a layout. Their optimization method is boosted by evaluating the STI effect on circuit performance. Although this work introduced several novel strategies for STI consideration, it would be more interesting if some other LDE aspects (such as WPE etc.) could be included as well.

Recently a new analog placement method was proposed in [57] that considered both LDE and analog constraints in the analytical formulation. In particular, three major sources of LDEs, including WPE, LOD, and oxide-to-oxide spacing, were transformed into nonlinear analytical placement models. Then an LDE-aware network-flow-based placement algorithm was utilized to mitigate the influence of the LDEs while improving circuit performance. For the matching devices, El-Kenawy and Dessouky [58] proposed a stress-aware pattern generation scheme that could utilize a mismatch factor to select the best pattern among candidates with the least current mismatch. Although in this work current mismatch due to STI is minimized, it would be more practical for analog circuits if WPE and oxide-to-oxide spacing could be also considered within the proposed mismatch factor. Moreover, a case study in [59] shows how STI and WPE can affect current mirror performance. Dummy structures were recommended to mitigate the LDEs in the analog circuits at the expense of silicon area usage. As expressed by the authors for their future work, an optimization method is still needed to address LDEs while conserving the redundancies.

Many previous works have tried to include OPC in their physical design process. The rule-based OPC has been widely deployed by numerous researchers. The concept of the rule-based OPC was initiated as early as the beginning of the century to address the issues related to the gate bridge and contact critical dimension variation for the digital circuit design in the sub-quarter micrometer technologies [60]. Li *et al.* proposed an intelligent OPC technique for process distortion compensation of layout mask [61]. Two different trials, first genetic algorithm (GA) with rule-based OPC and then GA with model-based OPC, were examined for the digital layouts. Due to the time-consuming nature of GA, the proposed methods strongly suffer from low computation efficiency. Most recently, Dong and Zhang proposed a rule-based OPC method for

analog layout migration [62]. This is among the first OPC works that are dedicated to analog layout optimization. Nevertheless, the proposed OPC-related handling is managed as a post-processing stage in the complex layout retargeting process.

Banerjee *et al.* [63] used a linear programming (LP) approach for model-based OPC to improve the runtime as well as mask complexity. Different from the conventional OPC tools, this work was aimed to reduce the error of electrical discrepancy between the ideal and the printed contours as a more reasonable objective for the low-k1 lithography, especially at 45nm and beyond. Since this work is mainly focused on the digital circuits by using current-matching to improve timing accuracy, it is unclear whether the proposed schemes can be still effective to the analog circuits that are often constrained by the contradicting multiple specifications. A 2D-wavelet-transform-based OPC model was developed in [64] to speed up the lithography simulation. Different from the popular coherent decomposition techniques, this work utilized an approximate simulation technique and proposed numerous optimization methods for improving polygon edge detection. The reported benefits include better scalability for large digital circuits and over 10X increase in runtime but with an average 5% error in terms of accuracy.

As a matter of fact, OPC can be included within routing algorithms. Over the years, OPC-aware routing has become an interesting topic in the EDA domain since the routing results can directly affect the quality of the target optical-proximity-corrected (OPCed) layouts. A routed layout with too many hot spots or critical paths may not be readily improved by OPC, which may otherwise significantly increase the mask complexity. Huang and Wong proposed an OPC-friendly maze router to solve the formulated multiple constrained shortest path problem based on the Lagrangian relaxation method [26]. By utilizing the symmetrical property of the optical systems, the light diffraction was efficiently calculated and stored in the look-up tables, from

which the costs that guided the router to minimize the optical interferences were obtained. Although effective, the proposed router was only implemented on two routing layers for experiments in the digital circuit design.

Mitra *et al.* [65] proposed a detailed routing that considered edge placement error (EPE) map as guidance to address the critical OPC areas. Since they utilized blockage to avoid routing in hot spots, this proposed algorithm may run out of layer resources, which contributes to higher amount of rip-up and reroute or even unroutability in the congested critical areas. Fang *et al.* [66] focused on the OPC cost modeling to help sequential maze router for better post-layout correction. In their new modeling, both spacing and routing patterns were considered within the routing stage. It would be more interesting if this work could be extended to a concurrent framework so as to be able to handle more constraints (e.g., analog constraints) simultaneously.

Most recently, capacitor array routing has become an interesting topic in the area of analog EDA since the capacitor arrays are often required as part of analog circuits (e.g., analog-to-digital converters (ADC) and digital-to-analog converters (DAC)). Each large capacitor in the analog circuits can be divided into a set of unit smaller capacitors, which are arranged in a rectangular common-centroid array. To achieve the least silicon area consumption, metal-oxide-metal (MOM) capacitors require new layout methods with routing even before placement. Chou *et al.* [67] proposed a routing and placement method, which was able to minimize the parasitic mismatch due to the routing among the capacitor arrays. In [68], a length-ratio matching routing was proposed by transforming the original problem into a wirelength minimization one. A two-step algorithm was developed to tackle the topology generation first and then detailed routing. Moreover, Lin *et al.* extended such an idea to form another recent work [69]. A novel capacitor sizing process for unit capacitors was introduced along with shielding routing and placement to

generate matched capacitor arrays while parasitics were considered during those operations. In summary, this nonconventional research subject has largely diversified the physical design research in the area of analog EDA.

2.3. Summary

In this chapter, we have first discussed the main concepts of three analog/RF challenges (i.e., EM, LDE, and OPC) in the advanced technology era. Since those issues have to be addressed before the real manufacturing takes place, it is the responsibility of the EDA tools to ensure they are well considered in the analog/RF design stage so as to make the layouts free from any failure or performance degradation before the final tape-out. Moreover, the previous works related to the abovementioned issues have been also reviewed with their advantages and drawbacks explained accordingly.

In the next chapters, our proposed adaptive ILP-based router will be described in detail. It will not only address the efficiency challenge which the previous LP-based algorithms often suffer from, but also tackle the performance-related issues for the analog/RF layout designs.

Chapter 3 Fast Hierarchical Adaptive Analog Routing Algorithm based on Integer Linear Programming

3.1. Introduction

The priority of analog constraints may affect the routing quality, which may in turn degrade the circuit performance if not being handled appropriately. Therefore, simultaneously satisfying multiple constraints for analog and RF circuits demands a generic optimization methodology, such as an ILP-based routing algorithm [31], which can not only balance multiple complex analog constraints, but also offer sound potential of integrating any emerging requirements called upon from the advanced submicron or nanometer technologies into the problem formulation/modeling. On the other hand, regular ILP-based routers normally suffer from a large number of variables and constraints, which would lead to high computational complexity and extensive running time. In the literature review above, we find that this deficiency problem has not been addressed by any of the previous studies. Therefore, in this work we are motivated to develop a general-purpose analog router, which is ideally able to simultaneously consider any existing and emerging analog constraints but still feature great advantage of computation efficiency [70].

In this chapter, we propose a heterogeneous hierarchical ILP-based router that can support adaptive routing resolutions at different levels of hierarchical framework while still maintaining good compatibility with complex analog constraints. Compared to the uniform hierarchical methods, our proposed heterogeneous hierarchical framework can create smaller regions at each

hierarchy level in order to make the ILP solving faster. A higher routing grid resolution is deployed for more congested areas, whereas a lower routing grid is applied to less crowded regions. Consequently, the number of variables and constraints decreases and the circuits can be routed much faster than the typical routing schemes. The unique contributions of this piece of work are summarized as follows:

- Our proposed non-uniform hierarchical routing method, which is initially based on uniform grids, is comprehensive and able to function automatically for any types of analog/RF circuits.
- The routing resolution at each hierarchy level can be optimally adapted according to the congestion status of the routing regions. Thus, the ILP-based routing efficiency can be significantly improved.
- Our proposed ILP modeling has better analog constraint coverage compared to the previously published works.
- It is possible to include other analog constraints, such as varied interconnect width, mismatch minimization, and electromigration, into the proposed router.

3.2. ILP-Based Analog Router

3.2.1. ILP Formulation

The basic formulation of our ILP-based routing will be described in the following three sections in terms of ILP variables, objective function, and constraints. Assume w, l and h represent the total width, total length, and total height of graph $G=(V,E)$. Set V can be defined as

$V=\{v_{i,j,k} |\ i \in W,\ j \in L,\ k \in H\}$, where $W=\{0, \ldots, w\text{-}1\}$, $L=\{0, \ldots, l\text{-}1\}$ and $H=\{0, \ldots, h\text{-}1\}$. For simplicity, we use notation v instead of (i,j,k) for index in the formulation expressions below.

In this chapter we use 45-degree octagonal routing grids [31], which mean that any vertices of graph G can be connected to their eight neighbors horizontally, vertically or diagonally with set $S=\{0,1,2,3,4,5,6,7\}$ used to index the segments as shown in Fig. 4(b). Set E, described as $E=\{e_{v,s} | v \in V,\ s \in S\}$, shows that each edge $e_{v,s}$ corresponding to one of the eight segments is to connect vertex v to one of its neighbors. Our ILP-based router first splits all the nets into *two-terminal nets*, each of which has only two ends corresponding to vertices in the graph. Then integer linear programs formulated for these two-terminal nets are further solved by using any ILP solver. In this work, we use FLUTE [37] to find the minimum-length Steiner tree and then two-terminal nets are extracted from the FLUTE output.

3.2.2. ILP Variables

Any ILP-based router is supposed to choose some edges in the graph to form paths for all the nets. Therefore, if for each net we define a variable set called R (standing for routing variables) associated with all the segments in the graph, the objective of "*finding the most suitable paths*" in the routing problem can be rephrased as "*setting variables in set R to the most suitable values*". Such a 0-1 ILP variable set is defined as $R=\{r_{v,s,n} | v \in V,\ s \in S,\ n \in N\}$, which means for any electrical net there are 8 associated variables (corresponding to segments) for any vertex $v_{i,j,k}$ within the graph. Here $v_{i,j,k}$ represents the position of the vertex in the three-dimensional graph where pair (i,j) is the two-dimensional planar location of the vertex and k is the actual layer assigned to the vertex. According to the R definition, the number of variables in the routing

region can be determined as $O(|N|\times|G|)$, where $|N|$ is the number of nets and $|G|$ is the graph size of $w\times l\times h$.

As defined above, set E includes all the edges on the layers but not between the layers. The edges, which connect the layers in between, are actually called *vias*, which are formulated as $VIA=\{via_{v,n}|v\in V, n\in N\}$. That is to say, one via is dependent on its vertex v and net n. Similarly, it can be modeled as a 0-1 variable, where 1 means there is a via in location (i,j,k) for net $n\in N$, otherwise it is equal to 0. Below we discuss how to express terminal and non-terminal sets. Assume $TERMINAL(n)$ is the set of terminal vertices for a net $n\in N$. The list of all the terminals can be described as $T=\{t_{v,n}|v\in TERMINAL(n), n\in N\}$ and thus the vertices not on this list are called *non-terminals*. In this thesis, the term of terminals also includes both ends of any two-terminal nets. A terminal, which is shared by multiple two-terminal nets belonging to one physical electrical net, is called *junction*. Note that analog/RF devices might have terminals on any metal layer. In addition, symmetric nets are listed in the paired set $N_{sym}=\{(n_i,n_j)|\ n_i,n_j\in N\}$ and the corresponding symmetry axes are marked as X_{n_i,n_j} (Y_{n_i,n_j}) for vertical (horizontal) lines.

3.2.3. Objective Function

Assume that $WL(n)$, $BN(n)$, $VN(n)$ and $CN(n)$ denote the wirelength, bend number, via number, and coupling noise of net $n\in N$, respectively. The objective of our ILP-based router is to minimize their total amount as below:

Minimize

$$\sum_{n\in N}\left(\rho_n^{WL}WL(n)+\rho_n^{VN}VN(n)+\rho_n^{BN}BN(n)+\rho_n^{CN}CN(n)\right), \tag{7}$$

where user-defined coefficients ρ_n^{WL}, ρ_n^{VN}, ρ_n^{BN} and ρ_n^{CN} describe the electrical net priorities in terms of wirelength, via number, bend number, and coupling noise. Since sensitive electrical nets may have higher priority than any regular ones in the analog and RF circuits, these coefficients can effectively help improve the circuit performance.

In more detail, all the variables in *R* belonging to net *n*, which are equal to 1 (i.e., those variables are the selected segments in the routing graph), would be accumulated to calculate $WL(n)$ as follows:

$$WL(n)=\sum_{k\in H}\sum_{v\in V}\sum_{s\in S}\left(r_{v,s,n}*\left(\rho_k*\omega_s*Swl\left(e_{v,s}\right)\right)\right), \tag{8}$$

where *r* belongs to *R* (i.e., $r\in R$ unless otherwise stated throughout the thesis) and $Swl(e_{v,s})$ is the wirelength of segment *s* for vertex *v*. As an example, $Swl(e_{v,4})$ is shown in Fig. 4(b) for vertex *v*, while $r_{v,s,n}$ can only be 1 (i.e., such a routing resource located at vertex *v* with segment *s* for net *n* is included in the final routing solution) or 0 (i.e., such a routing resource is not included in the final routing solution). User-defined coefficient ρ_k is used to express the layer priority coefficient of layer *k*, which means that certain layers may be preferred for routing in our proposed ILP modeling.

User-defined coefficient ω_s is used to denote the weight of segment *s*. This means that the designers can set different priorities for distinct segments by assigning proper values to ω_s. For instance, it might be useful when analog/RF designers are more interested in using 90-degree orthogonal routing grids for higher computational efficiency than the 45-degree octagonal ones.

In such a situation, more weight can be assigned to ω_s for *s* equal to 1,3,5, or 7 (i.e., a diagonal segment as shown in Fig. 4(b)) than the other orthogonal segments. This setting would eventually discourage the octagonal interconnection in the final routing solution since selecting diagonal segments is apparently more costly.

The via number is computed for non-terminals and terminals separately as described in (9),

$$VN(n) = VN_{NT}(n) + VN_T(n) \ , \tag{9}$$

where $VN_{NT}(n)$ and $VN_T(n)$ are the numbers of vias for non-terminals and terminals, respectively. For non-terminals, $VN_{NT}(n)$ can be counted by checking the segments present on different layers for all the vertices with the following logical AND (&) operation:

$$VN_{NT}(n) = \sum_{\substack{v1 \in V \\ v1 \notin T}} \sum_{\substack{k2 \in H \\ k2 > k1}} \sum_{s1 \in S} \sum_{s2 \in S} (k2 - k1) * \left(r_{v1,s1,n} \,\&\, r_{v2,s2,n} \right) \ , \ \ v1 = v_{i,j,k1}, \ \ v2 = v_{i,j,k2} \ , \tag{10}$$

where $(r_{v1,s1,n})$ represents segment *s1* of net *n* for vertex $v_{i,j,k1}$ at location *(i,j,k1)* in the graph. When different segments on distinct layers for one vertex, which can be measured by the logical operation above, have been selected by the ILP router, a via is ensured to be present there. So the via number should be incremented by one.

For terminals, the via number can be computed by:

$$VN_T(n) = \sum_{v1 \in T} \sum_{\substack{k2 \in H \\ k2 \neq k1}} \sum_{s2 \in S} |k2 - k1| * \left(r_{v2,s2,n} \right) \ , \ \ v1 = v_{i,j,k1}, \ \ v2 = v_{i,j,k2} \ , \tag{11}$$

where similarly *v1* (*v2*) is the abbreviation of $v_{i,j,k1}$ ($v_{i,j,k2}$). Equation (11) means that if a segment linked to terminal *v1* is not located on the terminal layer, there must be a corresponding via for *v1*. This equation ensures that ILP will respect the layer configuration of the terminals in any constituent devices, which might be on any metal layers.

The bends are enumerated by counting the number of vertices with two segments, which are not routed in the same direction as follows:

$$BN\ (n)=\sum_{v1\in V}\sum_{\substack{k1\in H\\ k2\geq k1}}\sum_{s1\in S}\sum_{\substack{s2\in S,s2>0\\ s2=(s1\pm pr)\%\ 8}}\left(r_{v1,s1,n}\ \&\ r_{v2,s2,n}\right),\ \ v1=v_{i,j,k1},\ \ v2=v_{i,j,k2}\ ,\tag{12}$$

where symbol *v1* (*v2*) is used to represent $v_{i,j,k1}$ ($v_{i,j,k2}$), *pr* is either 2 or 3 for representing the possible segment number range that might generate a bend on *v1*, and *%* is the remainder operator. In more detail, the correspondence between *s1* and *s2* in (12) can be described in Table 1.

Table 1. Correspondence between *s1* and *s2* in (12)

s1	0	1	2	3	4	5	6	7
s2	2,3,5,6	3,4,6,7	4,5,7,0	5,6,0,1	6,7,1,2	7,0,2,3	0,1,3,4	1,2,4,5

Coupling noise effect between two wires would be zero if two wires are shielded by another wire. Otherwise, it is calculated as follows:

$$CN(n)=\sum_{\substack{n1\in N\\ n1\neq n}}\sum_{d\leq Md}CP\left(e_{v1,s,n}\ ,\ e_{v2,s,n1}\right)*\left(r_{v1,s,n}\ \&\ r_{v2,s,n1}\right),\ \ v1=v_{i,j,k},\ \ v2=v_{i\pm d,j,k}\ ,\tag{13}$$

where symbol *v1* (*v2*) is used to represent $v_{i,j,k}$ ($v_{i\pm d,j,k}$), *Md* is the maximum distance of two segments with coupling noise effect, and $CP(e_1,e_2)$ is the function of calculating the coupling noise value between two edges e_1 and e_2 in the same metal layer. Although (13) above is only for calculating the horizontal coupling effect, the vertical one can be easily modeled on *j* rather than *i* in a similar way.

3.2.4. Constraints

In this Section, we only focus on the illustration of symmetry and routability constraints, while any other analog constraints (e.g., topology and length matching) proposed in [31] are not repeated here although they have also been implemented in our ILP-based analog router for experimental testing. For the symmetry constraints, the wire segments should be symmetrically routed with reference to the symmetry axis. The vertical symmetry can be expressed as follows:

$$r_{v1,s1,n1} - r_{v2,s2,n2} = 0, \;\; s1 \in \{0,1,2,3\}, \;\; s2 = (8 - s1)\%8 \;\; , \tag{14}$$

where

$$r_{v1,s,n1}, r_{v2,s,n2} \in R, \;\; (n1, n2) \in N_{sym} \;\; , \;\; i_2 = (2 * X_{n1,n2} - i_1), \;\; v1 = v_{i_1,j,k}, \;\; v2 = v_{i_2,j,k} \; .$$

Fig. 5 shows an example of symmetry constraint for two symmetric nets *Net1* and *Net2* with the reference of symmetry axis $X_{Net1,Net2}$ as shown in Fig. 5. Equation (14) guarantees that the router will choose the segments symmetrically, as can be seen in Fig. 5, with segment numbers 1 and 7 for symmetric vertices *S1* and *S2*, respectively.

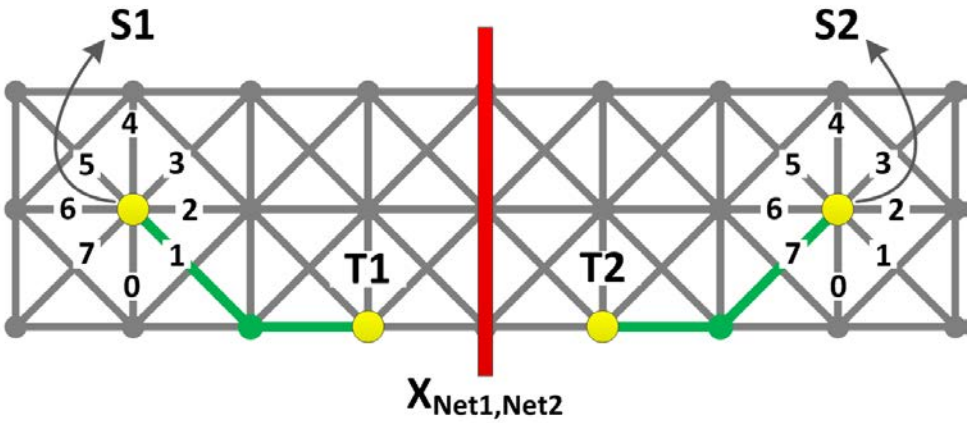


Fig. 5. Symmetry constraint of two vertices, *S1* and *S2* belonging to two symmetric nets *Net1* and *Net2* with symmetry axis $X_{Net1,Net2}$.

For the routability constraints, we have to make sure the following routing rules are followed:

- **Graph connectivity**:

  To maintain the connectivity of graph *G*, the variables of the neighbouring vertices are equal according to their relative locations. For example, since two vertices $v1=v_{i,j,k}$ and $v2=v_{i,j-1,k}$ are neighbours in the vertical direction, variables $r1=r_{v1,0,n}$ (that represents segments #0 of vertex *v1*) and $r2=r_{v2,4,n}$ (that represents segments #4 of vertex *v2*) are equal to ensure these two vertices are vertically connected.

- **Terminals**: Each terminal in *T* belonging to net $n \in N$ is connected to just one segment,

$$\left(\sum_{k \in H} \sum_{s \in S} r_{vx,s,n} \right) = 1, \quad vx = v_{i,j,k} \in T \quad . \tag{15}$$

- **Non-terminals**: Non-terminals of net $n \in N$ have to be connected to just zero or two segments:

$$\left(\sum_{k \in H} \sum_{s \in S} r_{vx,s,n} \right) = 0 \ \ OR \ \ 2, \quad vx = v_{i,j,k} \notin T \quad . \tag{16}$$

- **Layer overlapping**: This constraint as expressed below guarantees that the segment indices on different layers for the identical net are not the same, which can in turn ensure no occurrence of layer overlapping for any net $n \in N$:

$$\left(\sum_{k \in H} r_{vx,s,n} \right) \leq 1 \ , \quad vx = v_{i,j,k} \ . \tag{17}$$

- **Short circuit**: This constraint is to restrict that each vertex being a non-terminal can only be occupied by one net. The short circuit constraint is described in the following Equation (18).

$$\sum_{n \in N} \left(\sum_{s \in S} (r_{v1,s,n}) + via_{vx,n} \right) \leq 2 \,. \tag{18}$$

As defined in Section 3.2.2, *VIA* is the set of via variables and $via_{vx,n}$ (which is a member of this set) would be equal to zero if there is no via located at vertex *vx* in the routing graph belonging to net *n*, otherwise one. In our proposed ILP, it is allowed to have some terminals, which are located at one location *(i,j)* within the graph but on the different layers. This type of terminals is called *Conflict Terminals*, which will be described below in more detail.

- **Conflict Terminals**: Conflict terminals are defined as the terminals on the boundary of routing area, which are shared among different nets. The conflict terminals are normally discouraged to promote the routability. As an acceptable solution to routing conflict terminals, different nets should use distinct layers. Thus, the constraint can be described as follows:

$$\left(\sum_{n \in CFL} \sum_{s \in S} r_{vx,s,n} \right) \leq 1, \;\; vx = v_{i,j,k} \;\; , \;\; v_{i,j} \in conflict(CFL), \;\; k \in H \;\; , \tag{19}$$

where *CFL* is the set of netlists that share the conflict terminals, while function *conflict(CFL)* gets the *CFL* as input and returns *(i,j)* locations of the conflict terminals corresponding to the *CFL* list members.

- **45-degree bend:** This constraint ensures that there is no acute corner with angle of 45 degree at any location *(i,j)* in the graph.

$$\sum_{\substack{s1 \in S \\ s2=(s1+1)\%\ 8}} \left(r_{v1,s1,n} \;\&\; r_{v2,s2,n} \right) \;=\; 0, \;\; v1 = v_{i,j,k1} \;\; , \;\; v2 = v_{i,j,k2}, \;\; k1 \in H, \;\; k2 \in H, \;\; r \in R \;\; . \tag{20}$$

- **Crossing-segment short circuit**: This constraint ensures that there is no short circuit among different nets on their crossing segments. As can be seen in Fig. 4(a) and Fig. 4

(b), segment 3 of vertex $v_{i,j,k}$ has a crossing overlap with segment 5 of vertex $v_{(i+1),j,k}$. The constraint as expressed below would avoid choosing those two crossing segments for different nets:

$$\sum_{n \in N} \left(r_{v1,s1,n} + r_{v2,s2,n} \right) \leq 1, \quad s1 = 3, \quad s2 = 5, \quad v1 = v_{i,j,k}, \quad v2 = v_{i+1,j,k}, \quad i \in \mathrm{W}, \quad j \in L, \quad k \in H, \quad r \in R \ . \tag{21}$$

3.3. Adaptive Hierarchical Routing

Normally the ILP-based routing cannot efficiently route the entire circuit due to capability limitation of the ILP solver for big problems. Therefore, most of the previous works [31] [36] divide the routing area into smaller regions, which can be routed by ILP separately, one after another. Afterwards a set of solutions for those regions are concatenated to derive the entire routing result. However, a straightforward divide-and-conquer scheme may not be able to secure a high efficiency although it helps turn the entire problem to be solvable. In this work, we are focused on how to improve the computation efficiency of our ILP-based analog router without compromising the capability of meeting the special analog constraints.

3.3.1. Construction of Routing Regions

In this thesis, we also follow the concept of hierarchical routing to reduce the runtime. But in contrast to the previous work, we have developed a scheme of decreasing resolution in the hierarchical routing regions. Firstly the entire routing area is divided into multiple uniform regions, from which non-uniform regions are identified based on the location of the terminals settled in each region. Thus, the first level of our hierarchical routing includes some non-uniform

regions with different sizes. Compared to the initial uniform divisions, those non-uniform regions are normally smaller and thus can be routed faster. ILP can derive the routing solutions for the nets that are located at the first hierarchy level.

Algorithm 1. Routing region construction and two-terminal net generation

Input: *RgA* = (*RgA_X* × *RgA_Y*) /* routing area */
T /* set of net terminals */
RgS = (*RgS_X* × *RgS_Y*) /* size of uniform regions */

Output: *TT_N* /* set of two-terminal nets inside each region*/
Routing regions at each level

```
1   Call FLUTE to derive and store two-terminal nets to TT_N based on T;
2   Lv = 0;  /* hierarchy level initialization*/
3   While (RgS < RgA)  /* for each level*/
4     For (Rg_x = 0 to RgA_X with step size of RgS_X)
5       For (Rg_y = 0 to RgA_Y with step size of RgS_Y)
6         Use Rg_x and Rg_y to form the current routing region CRg;
7         Find the smallest bounding box (i.e., non-uniform region) inside CRg that covers all the
            inclusive terminals;
8         Treat the non-uniform regions from the previous hierarchy levels as blockage;
9         If (CRg contains a symmetric net)
10          Modify the non-uniform region to be equivalent to its symmetric counterpart and add them
              to the appropriate symmetry list;
11          Find and save all the pins of CRg symmetrically;
12        Else
13          Find and save all the pins of CRg;
14        End If
15        Find and save all the two-terminal nets inside CRg;
16      End For
17    End For
18    Prepare two-terminal nets based on the pins for the next level;
19    RgS_X = RgS_X × expansion_factor_X; /* expanding RgS_X in X direction*/
20    RgS_Y = RgS_Y × expansion_factor_Y; /* expanding RgS_Y in Y direction*/
21    Lv = Lv + 1;
22  End While
```

Algorithm 1 shows how routing regions are constructed and two-terminal nets are formed with our proposed scheme. We use FLUTE [37] to find the minimum-length Steiner tree and then two-terminal nets are extracted from the FLUTE output as described in Line 1. At each level *Lv*, bounding boxes of routing regions and intermediate pins are first identified. A bounding box is the biggest rectangle inside a uniform region that covers all the inclusive terminals. After finding the bounding box within the current region *CRg*, all the two-terminal nets that have exactly one terminal within *CRg* will be checked. The intersection between such two-terminal nets and the *CRg* boundary is called *intermediate pins* or simply *pins* (as shown in Line 11 or 13). Normally each hierarchy level would divide one long two-terminal net in *TT_N* into two types of subnets: one is located inside the current lower hierarchy level, while the other type of subnets is located at the upper hierarchy level and aimed to connect the pins between the boundaries of the current lower hierarchy level. Lines 10-11 show the handling of symmetric nets including updating symmetry list and symmetrically identifying pins. After that, the routing regions will be expanded by using two factors (*expansion_factor_X* and *expansion_factor_Y*) as described in Lines 19-20. In our implementation we set both factors as 2. It is worth pointing out that the bounding boxes of the routing regions are heterogeneous from one another as per the distinct locations of the enclosed terminals. In the next step, the regions at each hierarchy level constructed by Algorithm 1 would be sent to the ILP solver one level after another, where the routing regions of level *Lv* will be treated as blockage during the process of the next level (*Lv+1*) routing.

Fig. 6 shows one example with three nets (including a pair of symmetric nets *ABC* and *A'B'C'*) and their corresponding subnets. Dashed lines indicate the uniform routing region grid for forming the first hierarchical level. In Fig. 6, the whole routing area is divided into 8 uniform

regions by dashed lines. Based on the horizontal index (i.e., *Rg_x*) and vertical index (i.e., *Rg_y*), the uniform routing regions can be referred to as shown in Line 6 of Algorithm 1. For instance, the left-bottom panel can be expressed as region(0,0), while the right-top panel can be called as region(3,1). The red blocks marked as *Rg1-Rg7* correspond to the first hierarchy-level non-uniform routing regions. After identifying the routing regions at the first hierarchy level, the nets are divided into several subnets based on the pins found on the boundaries of the regions. As a result of the first-hierarchy-level routing region identification, there will be a list of non-uniform routing regions as output, each of which has a number of subnets to be routed.

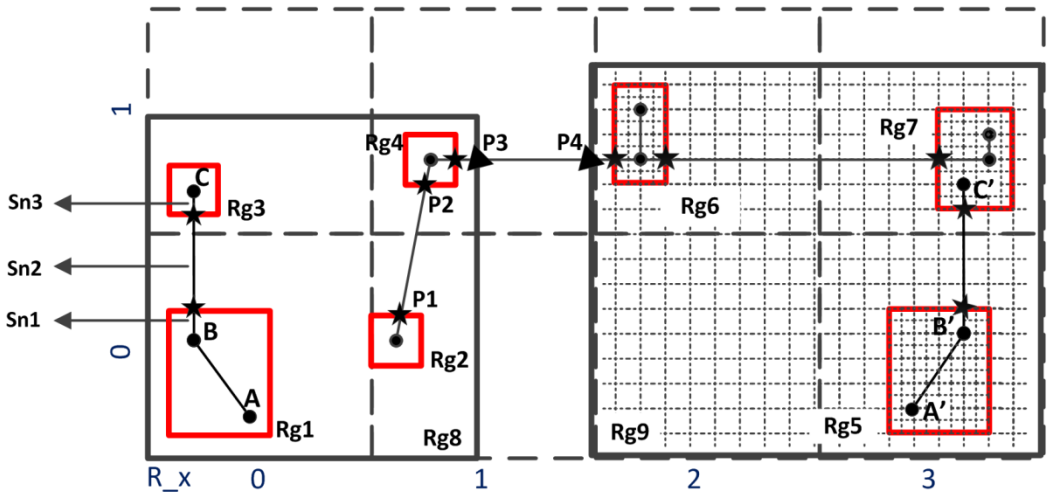


Fig. 6. Routing region construction at different hierarchy levels with intermediate pins on the boundaries.

As depicted in Fig. 6, the star-shaped points (e.g., *P1* and *P2*) are the pins for generating the subnets. For example, subnets *sn1* and *sn3* would be routed within regions *Rg1* and *Rg3*, respectively. As described in Lines 19 and 20 of Algorithm 1, every 2^i (where i is the number of hierarchy level) uniform routing regions would be combined to form one uniform routing region at the next higher hierarchy level. For instance, in Fig. 6 the leftmost four uniform regions, i.e., region(0,0), region(1,0), region(0,1), and region(1,1), would be merged to construct one bigger-

size uniform region at the second hierarchy level, while a similar situation takes place for the rightmost four uniform regions. Then the non-uniform regions at the second hierarchy level, which are marked in black in Fig. 6 (named as *Rg8* and *Rg9*), are created. At this level, the first hierarchy-level regions marked in red are seen as blockage and the subnets located beyond the red regions (i.e. *Sn2*) but inside the black regions would be routed by the ILP solver at the second hierarchy level. Similar to the first level, the subnets are divided into smaller subnets by assigning pins on the boundaries of the regions. In Fig. 6, the second hierarchy-level boundary pins are marked with triangle shape (e.g., *P3* and *P4*). The same scenario is repeated for the next upper levels until the highest level can cover the whole circuit.

To handle symmetric nets, we select their corresponding pins on the boundaries symmetrically. Moreover, a region with symmetric nets should be routed simultaneously with its counterpart region to ensure the symmetry constraints are satisfied synergistically. Fig. 6 shows an example of creating the regions of symmetric nets (marked as *ABC* and *A'B'C'*) as well as the pins for the next upper routing hierarchy level. At each hierarchy level, any symmetric nets have a list of symmetric regions in pairs. When ILP is invoked to route the regions one by one, the symmetry list would be checked. If one routing region is a symmetric one, its counterpart region would be extracted to derive the symmetric solution by ILP simultaneously.

Based on our modeling discussed above, the variables (i.e., routing resources) for nets are created just within their bounding box instead of the whole routing area. The rationale is that most of the nets are routed within the areas not much bigger than their bounding-box. And a bigger region than bounding-box might be utilized if there is an obstacle between the terminals. Based on this assumption, the number of routing variables should be at the level of $O(|N|\times|BB|)$, where $|BB|$ is the number of graph vertices within the bounding-box of all the associated nets.

3.3.2. Junction Handling

Junctions may be required for the nets that span multiple regions. One example is shown in Fig. 7(a), where *Rg1*, *Rg2* and *Rg3* are three regions at the first routing hierarchy level. *Net1* connects *S1* and *T1*, which are located in regions *Rg1* and *Rg3*, respectively. Region *Rg2* is located between *Rg1* and *Rg3*. Since these three regions would be seen as blockage at the second routing hierarchy level, *S1* and *T1* cannot be connected with minimum wirelength if they fail to be handled appropriately at the first routing hierarchy level. As shown in Fig. 7(a), one solution is to add a junction point inside *Rg2* to ensure that *Net1* will be routed properly. One junction point assignment scheme has been developed to make sure a suitable point would be selected. In a naive manner, adding one junction point in the center of a region is actually equivalent to generating two new two-terminal wires in that region (i.e., each two-terminal wire is to connect the junction point and one boundary intermediate pin). This is obviously inferior in terms of computation complexity. So in our proposed ILP-based router, we have deployed an alternative way of achieving better connectivity by adding just one two-terminal net in the region as shown in Fig. 7(b).

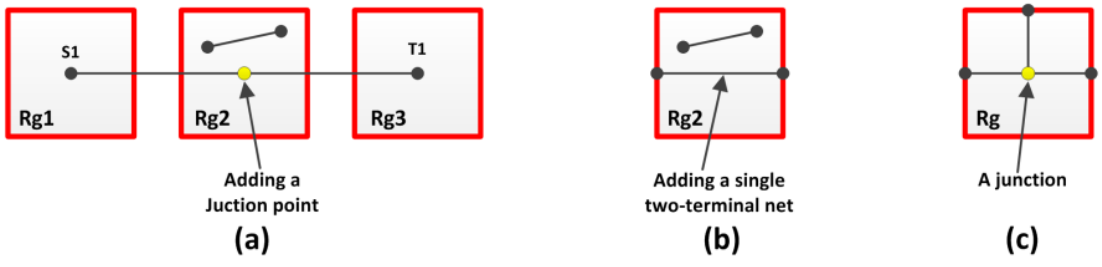


Fig. 7. (a) Inserting one junction point into intermediate region *Rg2*. (b) Replacing two new two-terminal wires with just one two-terminal wire. (c) A junction point connecting three two-terminal nets.

In addition, some regions may already enclose certain junctions, which are created by FLUTE or originally from the netlist. Such a junction actually serves as an intersection in a net to connect several two-terminal nets together. In our proposed ILP formulation, all of the nets inside a region are seen as two-terminal nets, no matter whether they belong to one physical electrical net or not. For those junctions, the short circuit constraint (as explained in Section 3.2.4) should not be imposed. Fig. 7(c) shows an example of a junction point shared by three two-terminal nets.

At the end of routing process at each hierarchy level, a list of un-routed regions will be reported by our ILP-based router and adjusted in the following refinement stage, where the troublesome regions would be ripped up and re-routed by reallocating junctions or pins on the routing region boundaries. Increasing the bounding box of certain nets or assigning more routing layers is also an effective way when such a limitation causes the routability problem. After that, the ILP solver will be invoked again to find the solutions for the uncompleted regions. After finishing the current routing hierarchy level in the adjustment stage, the data would be ready for the next upper routing hierarchy level.

3.4. Analog Routing Efficiency Enhancement

3.4.1. Variant Resolution in ILP Hierarchical Routing

One of the major differences between our hierarchical router and previous work is that we deploy variant routing resolutions among different routing hierarchy levels, that is, higher routing resolutions for lower hierarchy levels and lower routing resolutions for higher hierarchy

levels. In this way, the routing regions are divided into several variant resolution areas. The main reasoning of this scheme is that, in analog circuits, the heavily congested areas are located around the terminals of devices, which should be supplied with the highest routing resolution. However, the areas far away from the terminals are normally less crowded and thus may be routed with lower routing resolutions. Since lower resolution areas incur fewer vertices that in turn means fewer ILP variables, the whole routing problem can be solved much faster by using our proposed adaptive resolution scheme.

The resolution decrement ratio, which is the ratio between two applied routing resolutions at two consecutive hierarchy levels, is determined by Algorithm 2 that will be discussed in Section 3.4.2. To illustrate a general picture, in this Section we will first describe a simplified example that applies resolution decrement ratio of two for all the hierarchy levels. That is to say, for two consecutive hierarchy levels in this example, half resolution is always applied to the upper hierarchy level compared to the lower one. This halving-grid resolution scheme is demonstrated at the right half of Fig. 6, which includes routing regions *Rg5*, *Rg6*, *Rg7* and *Rg9*. In there regions *Rg5*, *Rg6*, and *Rg7* marked in red are located at the first hierarchy level, while region *Rg9* marked in black encloses the others and serves as the non-uniform region at the second hierarchy level. As can be seen, the routing resolution utilized for *Rg9* is halved compared to regions *Rg5*, *Rg6* and *Rg7*. Note that the boundary box of a non-uniform region as well as the associated pins at the lower hierarchy level should be always fitted into the immediately upper hierarchy level grids. In this way, all the routing regions can be nested seamlessly to construct the routing grids at the various hierarchy levels.

The number of the formed routing hierarchy levels can determine to what extent our proposed router is able to speed up the problem solving. In practice, the number of routing levels

depends on the size and topology of a circuit. Our experimental results show that around 3-4 hierarchy levels are needed for a regular analog building-block circuit (e.g., consisting of a dozen devices), and about 5-6 hierarchy levels are required for a fairly large analog circuit. To avoid such deficiency, we have developed an algorithm in Section 3.4.2, which can determine the optimal resolution decrement ratios based on the layout congestion status.

3.4.2. Dynamically Adjustable Resolution Decrement Ratio

Based on the discussion in the previous Section, we advocate adaptively decreasing routing resolutions for any two consecutive hierarchy levels. To improve the routing efficiency, we propose a new concept of dynamically adjustable resolution decrement ratio for analog/RF ILP-based routers. The resolution decrement ratio is dynamically adjusted according to the congestion level of the routing regions. That is to say, less congested layouts would have higher resolution decrement ratio compared to the more crowded ones. However, routing quality is another factor needed to be considered along with speed-up benefit from decreasing the routing resolutions. We have developed a branch-and-bound-based heuristic scheme, Algorithm 2 as shown below, in order to balance speed-up and routing quality by finding the most suitable resolution decrement ratios.

To be aligned with our routing grid hierarchical transformation, we constrain the possible resolution decrement ratios within a set of $RDR = \{2^i | i>0, i \text{ is integer}\}$ as shown in Line 4 of Algorithm 2. Before starting the ILP-based detailed routing along the hierarchical flow, routing quality needs to be estimated in the global routing stage. The routing quality (*RQ*) covers number of vias, number of bends, and wirelength at all the hierarchy levels as defined in (22):

$$RQ = \alpha \times ViaN + \beta \times BendN + \gamma \times WireLength \ , \quad (22)$$

where *ViaN* is the number of vias, *BendN* is the number of bends, and *WireLength* is the total interconnect length in the layout. And α, β and γ are the user-defined coefficients. For any routing region, these routing quality constituent parameters can be quickly estimated by searching two-terminal nets inside the region. For instance, *ViaN* can be estimated by computing the number of different nets, which are crossing each other in the region. *BendN* would be equal to either one or zero in each region for any two-terminal net. If there is a direct connection in terms of octagonal directions between the ends of one two-terminal net, *BendN* would be equal to zero; otherwise one. *WireLength* is equal to the Manhattan distance sum of all the two-terminal nets located in the layout.

Moreover, the applied resolution decrement ratio can be used for representing the speed-up measurement in the hierarchical routing. A larger resolution decrement ratio can lead to less runtime for better algorithmic efficiency. In addition, the routing quality is able to reflect the status of routing congestion since *ViaN*, *BendN* and *WireLength* would increase in the congested regions if a higher value of resolution ratio is selected from set *RDR*.

As shown in the pseudo code below, Algorithm 2 has the same input as Algorithm 1, which is called inside (in Line 1). Lines 1-9 is the initialization stage for the variable set. To calculate the routing quality, the estimation needs to run from the highest hierarchy level based on (22). Therefore, in Line 5-7 a group of resolution decrement ratios are initialized as the contents of array *BestRDR[]*, which is supposed to hold the best resolution decrement ratios identified thus far. Then from the highest hierarchy level to the lowest one, the algorithm keeps using the branch-and-bound technique to identify the largest resolution decrement ratio at each hierarchy level as shown in Lines 10-19.

Algorithm 2. Determination of optimal resolution decrement ratios

Input: $RgA = (RgA\_X \times RgA\_Y)$ /* routing area */
T /* set of net terminals */
$RgS = (RgS\_X \times RgS\_Y)$ /* size of uniform regions */

Output: Determined resolution decrement ratios for all the hierarchy levels

1	Call Algorithm 1 to derive the number of the maximum hierarchy level, which is stored in variable *LvMax*;
2	*Lv = 0*; /* variable representing the current hierarchy level */
3	*BestRq* = 0; /* variable representing the best routing quality up to now*/
4	*RDR* = {2^i|*i>0, i* is integer}; /* list of possible resolution decrement ratios*/
5	**For** (All hierarchical levels from *Lv* to *LvMax*)
6	*BestRDR*[*Lv*] = 2^{Lv}; /* initialization of array *BestRDR[]* storing the best combination of resolution decrement ratios at all the hierarchy levels*/
7	**End For**
8	Estimate the routing quality based on *BestRDR[]*, which is saved to *BestRq*;
9	*Lv = LvMax*;
10	**While** (*Lv* > 0)
11	**For** (All hierarchy levels from *Lv* to *LvMax*)
12	**For** (All possible values from *RDR* at level *Lv* that have not been evaluated yet)
13	Use the branch-and-bound technique to identify the largest resolution decrement ratio that can bring an acceptable routing quality with reference to *BestRq;*
14	Keep updating *BestRDR[]* and *BestRq*;
15	**End For**
16	Keep updating *BestRDR[]* and *BestRq*;
17	**End For**
18	*Lv = Lv* – 1;
19	**End While**
20	**Return** *BestRDR;*

For example, suppose the number of the maximum routing hierarchy level is equal to four. A possible resolution decrement ratio would be one value from set *RDR = {1, 2, 4, 8, 16, 32, 64…}*. After the loop of Lines 5-7, the initialized *BestRDR[]* would be 1, 2, 4 and 8 for the *Lv* levels of 0, 1, 2 and 3, respectively. And *BestRq* will be estimated accordingly for such a setting in *BestRDR[]*. Then the algorithm will check all other possible resolution decrement ratios. Since

ratio of 8 has been already evaluated for the highest hierarchy level of $Lv = 3$ in this example, any higher ratios, such as *16, 32, 64, etc*. from set RDR would be verified. For each higher ratio, if the associated routing quality is acceptable within a user-defined tolerance compared to *BestRq*, that higher ratio will be stored into *BestRDR[]* and the corresponding derived routing quality will be used to update *BestRq*. It is worth mentioning that, based on the branch-and-bound technique, the rest of higher ratios in set RDR would not need any further evaluation if one ratio was found infeasible. Once evaluation is completed for one level, the search would come back to the lower levels until reaching the lowest hierarchy level as can be seen in Line 10. Eventually the algorithm outputs the optimal combination of identified resolution decrement ratios stored in *BestRDR[]*.

The deficiency about the limited number of hierarchy levels as described in Section 3.4.1 can be effectively resolved by using our proposed dynamically adjustable resolution decrement ratios in Algorithm 2. In this way, although the number of total hierarchy levels is relatively small for certain uniformly sparse layouts with small or medium size, the actually applied resolution can be adapted to very low in order to hit the efficiency peak since a higher routing resolution is useless anyway. By using the same idea, in the next Section white space handling, which is also a dynamic decrement of resolution ratio, can be especially helpful for certain highly sparse regions to improve the routing efficiency of our proposed ILP-based analog/RF router.

3.4.3. White Space Handling

White space (WS) is defined as an area that has no terminals or pins inside. Since an analog/RF layout is normally much sparser than its digital counterpart, our proposed router can create more routing hierarchy levels for efficiency improvement if considering the white space regions. The pseudocode of the white space generation is listed in Algorithm 3.

Algorithm 3. White space generation

Input: *T* /* set of net terminals */

Output: *List_of_WS* /* list of white spaces */

```
1   Call FLUTE to derive and store two-terminal nets to TT_N based on input T;
2   Initialize List_of_WS as NULL;
3   For (any two-terminal net, called ST_N, in TT_N )
4      Derive the enclosing region (called Src_Rg) of ST_N's source terminal;
5      Derive the enclosing region (called Snk_Rg) of ST_N's sink terminal;
6      If (Src_Rg is not a neighbor for Snk_Rg)
7         Curr_WS = NULL;  /* current WS*/
8         For (any region (called Rg) from Src_Rg to Snk_Rg)
9            If (Rg is empty)   /* i.e., no terminal inside Rg */
10              Add Rg to Curr_WS;  /* enlarging Curr_WS to contain Rg*/
11           Else /* the current region is not a white space*/
12              Add one junction inside Rg; /* as described in Section 3.3.2*/
13              Add Curr_WS to List_of_WS and make Curr_WS empty;
14              Continue the white space search from the junction and its region;
15           End If
16        End For
17     End If
18  End For
19  For(any white space region (called Fst_WS) in List_of_WS)
20     For(any white space region (called Sec_WS, Sec_WS≠Fst_WS) in List_of_WS)
21        If (there is overlap between Fst_WS and Sec_WS)
22           Split Sec_WS into several non-overlapping white space blocks;
23           Remove Sec_WS from List_of_WS;
24           Add the new non-overlapping white space blocks into List_of_WS;
25     End If
26  End For
```

The white space generation algorithm takes a set of two-terminal nets computed from FLUTE as input and returns the list of white space regions as output. For any two-terminal net blocked by intermediate regions, the source/sink terminals and their enclosing regions are first derived (Lines 4-5). Then the routing regions are checked one after another. If there is no

terminal in the current region, it would be merged to *Curr_WS*. This means that all empty regions are merged together to ensure that the largest white space would be extracted. However, if there is a non-empty region among the swept intermediate regions, a junction is needed to ensure the connectivity as explained in Section 3.3.2. An example of white space handling is shown in Fig. 8, where a net connects terminal *S* in region *Rg1* to *T* in region *Rg6*. Based on Algorithm 3, regions *Rg2* and *Rg3* would be merged as an entire white space named *Ws1* (as shown by the dashed block in green color). Since region *Rg4* contains terminal *A*, it is a non-empty region and one junction *J* is added to this region. *Ws1* will be saved into *List_of_WS* as described in Line 13 of Algorithm 3. Then new white space exploration would be continued after replacing source terminal *S* with junction *J*. The algorithm will then find *Ws2* as the next white space.

In Lines 19-26 of Algorithm 3, the white space blocks inside *List_of_WS* will be checked for any overlap incidents. If there is an overlap, one white space block will be split into several non-overlapping ones. By processing the white space area, we can ensure that a maximum number of levels can be handled by our proposed hierarchical router to gain a boosted routing efficiency. As another application of dynamically adjustable resolution decrement ratio concept discussed in Section 3.4.2, the speed-up effectiveness of the white space handling scheme strongly depends on circuit congestion. The hierarchy level number for a white space region can be derived from (23):

$$Log_{Rd}\left(D_{ws} / D_{hr}\right) , \tag{23}$$

where D_{ws} is the applied segment length within the white space and D_{hr} is the highest resolution used in the hierarchical routing. One example of the segment length is shown in Fig. 4. *Rd* is the

resolution decrement ratio from level *l* to level *l+1*. For example, when *Rd* = 2, it means that the resolution for level *l+1* would be half of that for level *l*.

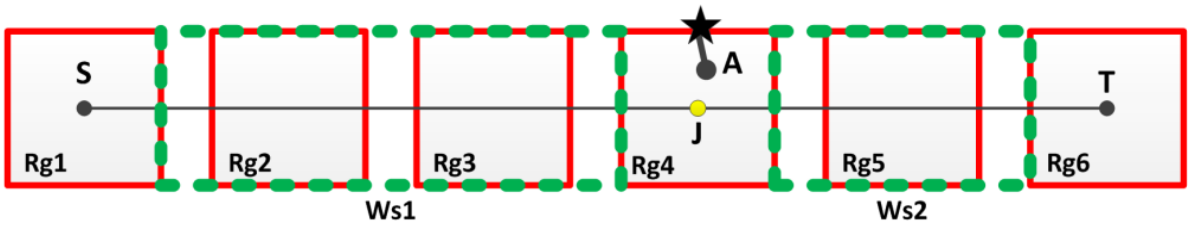


Fig. 8. An example of white space generation for one net connecting *S* and *T*.

3.5. Experimental Results

We implemented our proposed routing algorithm in C++ and the experiments were run on an Intel X86 1.2GHz Linux workstation that has 64GB of memory. We utilized Mosek 7 [71] for solving the ILP problems. Our experimental test circuits include one high-speed analog operational amplifier (OPAMP), one analog dynamic comparator, one differential low-noise amplifier (LNA), and one artificial analog circuit. The OPAMP and comparator feature the size of regular analog building-block circuits, while the LNA represents an example with regular RF building-block circuit size. The artificial circuit, enclosing 70 nets and 60 devices, is used to represent the group of large circuits in terms of device count and net number. It was built up by merging five analog dynamic comparators together into one circuit. Among them, four comparators are located very far from one another and the fifth one is a comparator which is enlarged in terms of transistor sizes and distances among the transistors. All of those five circuits have their own analog constraints as the features of the original comparator. The inputs, references, power, and ground nets are connected to one another to make the circuit very similar

to a real large and complicated analog circuit. The features of these experimental circuits are listed in Table 2. The schematics of the OPAMP, comparator and LNA are depicted in Fig. 9, Fig. 10 and Fig. 11 respectively.

Table 2. Features of four testbench circuits

Circuits	# Devices	# Nets	Area(um^2)
OPAMP	**10**	**10**	**1276**
Comparator	**12**	**14**	**208**
LNA	**14**	**16**	**1130360**
Artificial Circuit	**60**	**70**	**564570**

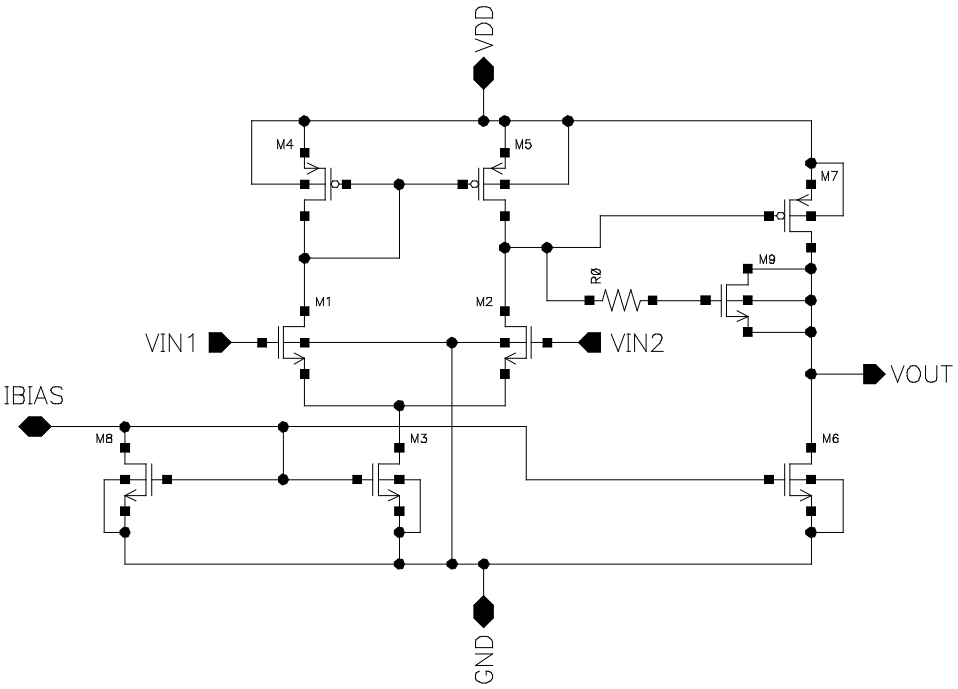


Fig. 9. Schematic of the OPAMP circuit.

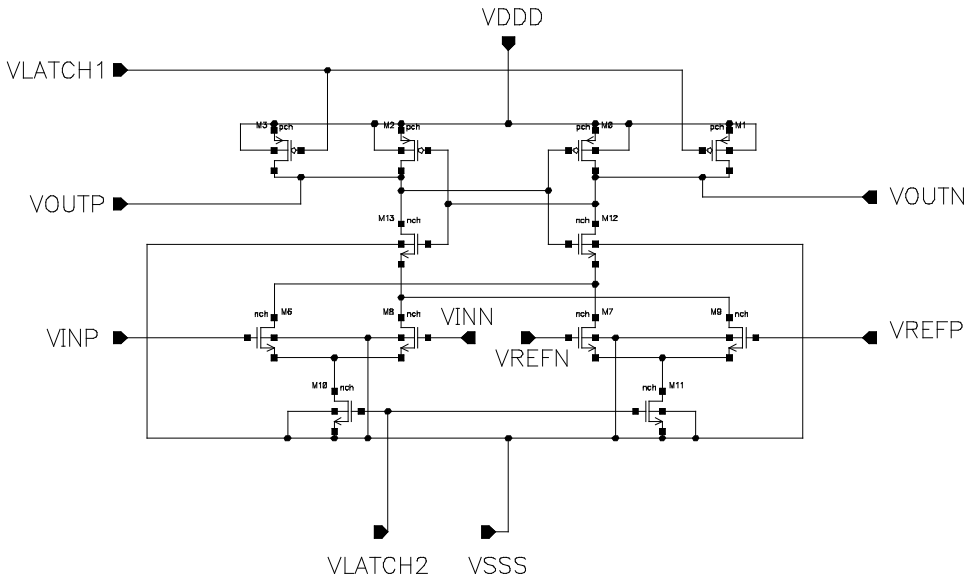


Fig. 10. Schematic of the comparator circuit.

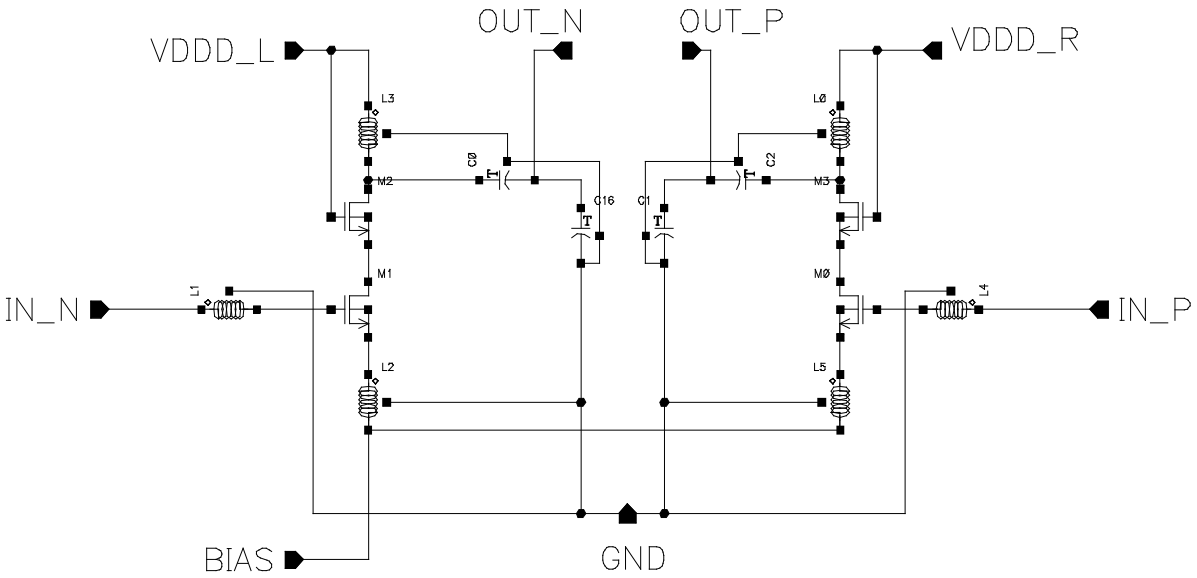


Fig. 11. Schematic of the LNA circuit.

Table 3 provides the comparison among different methods. We implemented the method from [31] as shown in the second column of Table 3. In column "*AG*", the adaptive grid resolutions of hierarchical routing are decreased (based on Algorithm 2 discussed in Section 3.4.2) from the lowest to highest hierarchy levels. In addition, term "*WS*" means the white space scheme as discussed in Section 3.4.3, which has also been included in the router. On

top of that, the column of "*AG+WS*" records the experimental results from our proposed router [72], which includes both adaptive resolution and white space handling for hierarchical routing. The data provided in Table 3 are shown in two columns, one for the absolute performance amount and the other for the percentage increment with reference to the method of [31] by using the following:

$$Percentage = \frac{Reference - Performance}{Reference} \times 100\% \ , \tag{24}$$

where *Performance* is the specific performance amount from one method and *Reference* is the corresponding amount from method [31]. Therefore, a positive percentage shows performance superiority, while a negative percentage indicates performance inferiority.

As can be seen in Table 3, the running times are reduced by around 12.5%, 34.9%, 87.9%, and 89% for the OPAMP, comparator, LNA, and artificial circuit, respectively, if using our proposed scheme *AG+WS* compared to method [31]. It is observed that a regular ILP-based router gets very slow when the circuit size turns bigger. But our proposed method is very efficient for the bigger circuits. In particular, the white space handling scheme can improve the running efficiency significantly for the LNA and the artificial circuits. The total numbers of variables and constraints utilized by the distinct methods are also reported for these test circuits. As can be seen in Table 3, our proposed *AG+WS* router reduces the number of variables by 16.6%, 38.6%, 74.6% and 86.7%, while it decrease the number of constraints by 4%, 30.9%, 69.2% and 81.8% respectively for four test circuits. In the meantime, our proposed AG+WS method experience a tiny increase in wirelength compared to method [31], by up to 2.4%, 1.4%, 0.4% and 0.5% for the OPAMP, comparator, LNA, and artificial circuit, respectively. On the other side, the number of vias (bends) is increased by 7.1% (19%), 12.5% (19.3%) and 19.3%

(14.8%) for the comparator, LNA, and artificial circuit, respectively, while there is no increase for via (bend) number in OPAMP circuit.

We have also experimentally studied the difference between 45-degree octagonal and 90-degree orthogonal routing grid modeling in terms of variable number and constraint number. In the orthogonal modeling, there is no diagonal connection between the vertices in the routing grid graph. It means that only the segments with index 0, 2, 4 and 6 (shown in Fig. 4(b)) are utilized to link the vertices in the graph. Apparently the orthogonal modeling may find inferior routing solutions due to only half options available compared to the octagonal modeling. Nevertheless, the orthogonal routing grid modeling can generate much fewer variables and constraints, which in turn makes it faster than the octagonal one. For instance, in our experiment 47,558 variables and 85,898 constraints were generated for ILP solving in the LNA circuit when orthogonal modeling was used whereas those numbers would soar to 136,122 and 341,142 in octagonal modeling as shown in Table 3. Therefore, it might be preferable for the orthogonal modeling if the performance such as wirelength is not that critical from the interest of designers for the sake of high computation efficiency.

Table 3. Comparison of four different circuits on various methods

Circuits		Method - [31]	AG		WS		AG+WS	
			Value	%	Value	%	Value	%
OPAMP	*Runtime (sec.)*	16	14	12.5	16	0	14	12.5
	Wirelength(um)	170	174	-2.4	173	-1.8	174	-2.4
	# of variables	9338	8406	10	8362	10.5	7786	16.6
	# of constraints	16307	15971	2.1	16149	1	15653	4
	# of Via	13	13	0	13	0	13	0
	# of Bend	18	19	-5.6	18	0	18	0
Comparator	*Runtime (sec.)*	83	69	16.9	64	22.9	54	34.9
	Wirelength (um)	137.1	142.3	-3.8	137.2	-0.1	139	-1.4
	# of variables	74108	62202	16.1	50508	31.8	45514	38.6
	# of constraints	160507	135820	15.4	124028	22.7	110878	30.9
	# of Via	28	36	-28.6	34	-21.4	30	-7.1
	# of Bend	42	54	-28.6	43	-2.4	50	-19
LNA	*Runtime (sec.)*	1590	616	61.3	253	84.1	192	87.9
	Wirelength(um)	9043	9445	-4.4	9040	0	9081	-0.4
	# of variables	536296	399434	25.5	186934	65.1	136122	74.6
	# of constraints	1107314	822658	25.7	494909	55.3	341142	69.2
	# of Via	16	22	-37.5	18	-12.5	18	-12.5
	# of Bend	88	122	-38.6	105	-19.3	105	-19.3
Artificial Circuit	*Runtime (sec.)*	6713	4935	26.5	1376	79.5	737	89
	Wirelength (um)	7417	7810.6	-5.3	7420.9	-0.1	7455.6	-0.5
	# of variables	4060120	3182990	21.6	699690	82.8	540506	86.7
	# of constraints	7723629	6307446	18.3	1846211	76.1	1403275	81.8
	# of Via	218	324	-48.6	276	-26.6	260	-19.3
	# of Bend	862	1148	-33.2	950	-10.2	990	-14.8

Since both the LNA and the artificial circuit can be handled more efficiently compared to the previous works, we can conclude that our proposed router is scalable for more complex and larger circuits. In terms of vias and bends, one can see that our proposed method may incur up to 19.3% increase compared to method [31] for all the testing circuits in our experimental settings. If necessary, this may be mitigated by assigning relatively high weights for the user-defined coefficients ρ_n^{VN} and ρ_n^{BN} in (7). The layouts of the routed OPAMP, comparator and LNA by our proposed ILP-based router are shown in Fig. 12, Fig. 13 and Fig. 14 respectively.

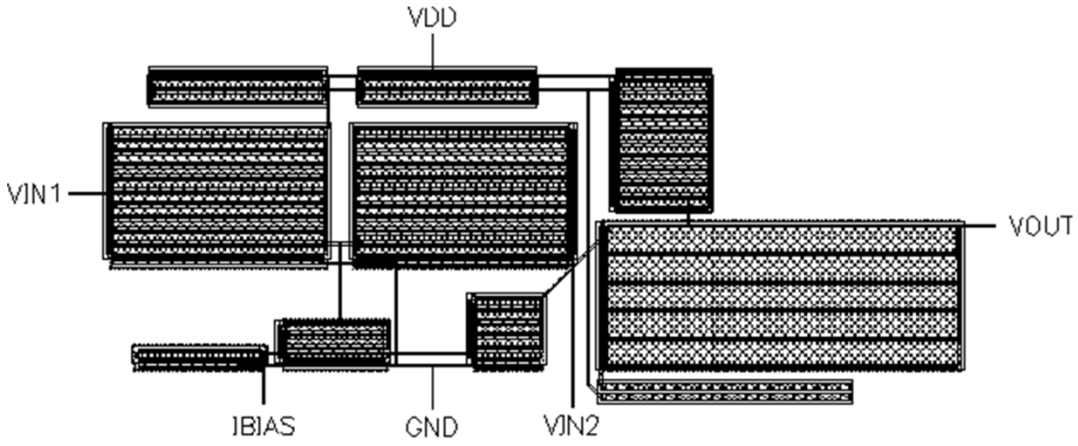


Fig. 12. OPAMP layout by using our proposed *AG+WS* method.

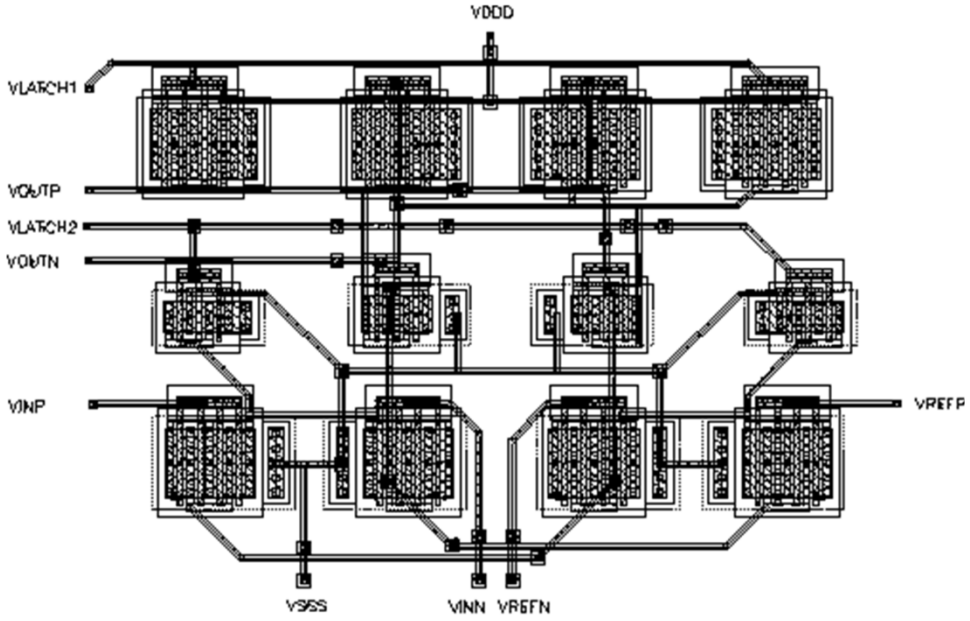


Fig. 13. Comparator layout by using our proposed *AG+WS* method.

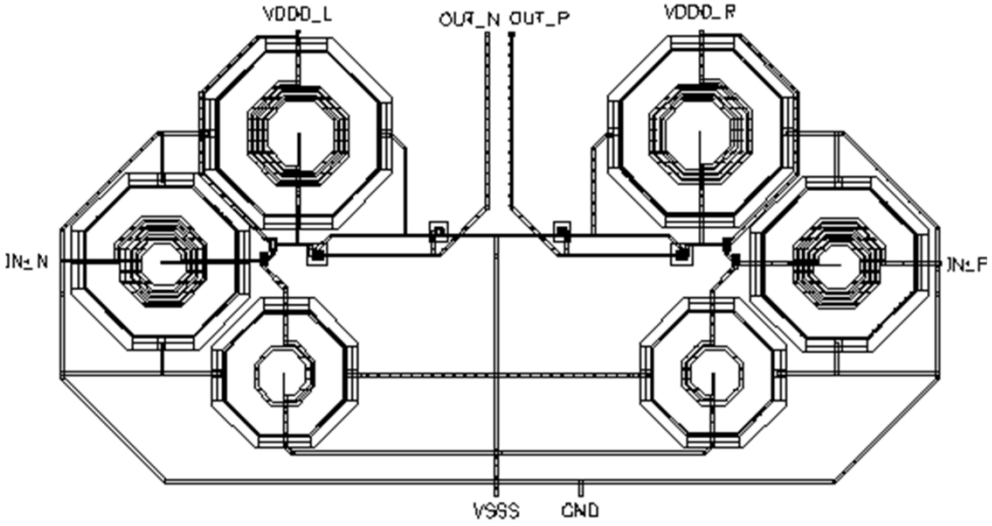


Fig. 14. LNA layout by using our proposed *AG+WS* method.

The schematic and post-layout simulation performances of the OPAMP, comparator and LNA are reported in Table 4. The design specifications and schematic performance are reported in the second and third columns. The circuit performance data from method [31] and our *AG+WS* router are given in the fourth and fifth columns respectively. The manual design from expert analog/RF layout designers is also included in the last column of the table for a comprehensive comparison with the other methods. As can be seen in this table, the final layout routed by our proposed method passed all the specifications of the comparator and LNA circuits. However, the layout routed by method [31] failed to pass the specifications of the comparator and LNA.

As described in Section 3.2.3, our ILP objective function takes into account layer priority and net priority, which are important factors in the analog and RF layout design. Since we impose more priorities to the output nets of the comparator circuit, the propagation delay and average power consumption improves 14.2% and 10.3%, respectively. According to our experiments, layer priority in LNA circuits can largely affect the performance of this sensitive

RF circuit. As can be seen in Table 4, our proposed router is able to pass the specification for S22 by properly handling the layer priority while the regular method used in [31] could not pass the S22 specification. Moreover, as can be seen from Table 4, the performances of our proposed method are very close to the manual ones generated by expert analog/RF layout designers.

Table 4. Schematic and layout performances of OPAMP, comparator and LNA circuits

Circuits		Spec.	Schematic	Method [31]	Our Proposed Method		Manual Design
					Value	%	
OPAMP	***DC Gain(dB)***	60	60.8	60.11	60.26	-0.2	60.28
	Gain Margin(dB)	10	22.86	10.5	10.23	2.6	10.45
	Phase Margin(degree)	60	65.75	61.23	60.42	1.3	60.94
	Unit Gain Band- width (MHz)	300	376.8	339.8	356.7	-5	356.5
Comparator	***Propagation Delay (ps)***	600	506.7	627	538.2	14.2	537.1
	Positive falling-edge overshoot (mV)	250	213.7	236.2	233.7	1.1	233.7
	Negative falling-edge overshoot (mV)	150	60.66	134.5	126.1	6.2	125.9
	Average power (uW)	12	9.95	11.7	10.5	10.3	10.5
LNA	***Noise Figure(dB)***	2.5	1.88	2.15	2.12	1.4	2.10
	Gain(dB)	16	18.24	16.56	16.88	-1.9	16.88
	S11(dB)	-10	-15.34	-18.08	-18.93	-4.7	-19.97
	S22(dB)	-10	-15.23	-8.8	-11.69	-32.8	-11.79
	IIP3(dB)	-7	-6.49	-6.29	-6.53	-3.8	-6.69

3.6. Summary

In this chapter, we have presented an adaptive hierarchical ILP-based routing algorithm for analog and RF integrated circuits. Besides promoting smart formulation of special analog constraints in ILP, we have developed a methodology of adaptively decreasing routing resolution in the hierarchical routing that can largely reduce the associated ILP variables and constraints. So the routing efficiency can be significantly boosted for the analog and RF circuits. In addition, a

white-space handling scheme has been developed to further improve the routing efficiency without compromising the routing quality, especially helpful for fairly larger analog and RF circuits. The experimental results demonstrate the efficacy of our proposed algorithm over the previously published works. In addition, the circuit performance has been improved compared to the previous work since our proposed routing model includes more analog and RF circuit performance-related constraints.

Chapter 4 Electromigration- and Parasitic-Aware ILP-Based Analog Router

4.1. Introduction

In this chapter, to tackle the performance challenges and deal with the analog/RF constraints simultaneously, we will propose an ILP-based concurrent router that can address electromigration, interconnect parasitics and mismatch minimization concerns all together along with the other analog/RF constraints. Electromigration has been considered as a type of new constraints in our ILP-based routing process. Interconnect parasitics are utilized in the sensitivity analysis to devise the most suitable interconnect widths that can lead to the best circuit performance. This means that our proposed router is aware of the interconnect parasitics by assigning proper wire/via width for each subnet. Another aspect of our proposed interconnect parasitic control algorithm is to minimize the mismatch effect if it cannot be completely eliminated by nature. In contrast, the conventional analog routing methods can neither allow for the overlapping scenario of the matching nets nor deal with the minimization of mismatch. To the best of our knowledge, this is the first EM- and interconnect-parasitic-aware concurrent analog router, which is able to address mismatch problems for the sensitive nets in the analog/RF circuits.

4.2. EM- and Parasitic-Aware Analog Routing Problem Definition

Since in this chapter we consider EM and parasitic in our router, the EM- and parasitic-aware analog routing problem (explained in Section 2.2.1) can be formally refined as "*finding the most suitable set of edges in graph G that can define paths among the terminals of the electrical nets besides satisfying the EM, routability, and other analog/RF constraints while interconnect parasitics are optimized and mismatch is minimized in light of the given specifications"*.

Within the routing graph *G*, the horizontal and vertical routing resolution is defined as the distances between two neighboring vertices in the horizontal and vertical directions, respectively. Such routing resolution is determined by the minimum distance defined in the process design rules for one specific technology. Since analog/RF designers or certain circumstances might need finer resolution than the default value for more accurate routing, there is a custom-resolution option in our routing graph construction stage that can enable higher resolution to be imposed on the routing graph *G*.

4.2.1. ILP-Based Analog Routing Formulation

Since the basic formulation of our ILP-based routing has been explained in Section 3.2, here we only present new formulation for symmetry, wire width, and via widening constraints. To include the mismatch minimization constraint, the symmetry constrain formulated in (14) will be later divided into two individual equations detailed in Section 4.3.5.

Wire width constraints are used to confine certain nets to reach specific widths. The following formulation can be utilized for the width of 2 (i.e., two units of routing grid) on net *nx* with reference to the basic routing grid, while a bigger width value can be modeled in a similar way:

$$\underbrace{\left(\sum_{n\in N}\left(\sum_{s\in S}(r_{v1,s,n})+via_{v1,n}\right)\right)}_{Term1}+\underbrace{\left(\sum_{s\in S1}r_{v2,s,nx}\right)}_{Term2}\le 2, \tag{25}$$

$$v1=v_{i,j,k}\ ,v2=v_{i,j-1,k}\ ,S1=\{1,2,5,6\}\ ,$$

$$\left(\sum_{n\in N}\left(\sum_{s\in S}(r_{v1,s,n})+via_{v1,n}\right)\right)+\left(\sum_{s\in S1}r_{v2,s,nx}\right)\le 2, \tag{26}$$

$$v1=v_{i,j,k}\ ,v2=v_{i-1,j,k}\ ,S1=\{0,3,4,7\}\ .$$

The following example shows how the width constraints can ensure a proper width for net *nx* as depicted in Fig. 15. Assume that *rx* belonging to net *nx* (marked in green) with segment number 2 (from set *S1*) in (25) has been selected into the routing path. Vertex *v1* located at *(i,j)* and vertex *v2* at *(i,j-1)* shown in Fig. 15 have the notations as defined in (25). As can be seen in Fig. 15, to make sure the width of *rx* is equal to 2, the upper segment called *rx′* should be reserved for net *nx* when *rx* has been selected in the routing path. This means that vertex *v1* (and therefore all of its segments that are marked in blue in Fig. 15) cannot be used as routing resources by the other nets except for *nx*. To explain (25) more clearly, we define *Term1* and *Term2* to distinguish two portions within (25). If *rx* is equal to one (which means *rx* is part of the routing path of net *nx*), *Term2* would be equal to two if we suppose that *v2* is a non-terminal (due to the non-terminal constraint). Thus, to satisfy the constraint in (25), *Term1* has to be zero, which means none of the other nets can use vertex *v1* in their routing path. In other words, if *rx* is equal to one, vertex *v2* would be belonging to net *nx*. Therefore, vertex *v1* needs to be reserved for net *nx* accordingly. On the other hand, the constraint in (25) guarantees that the other nets can include vertex *v1* in their routing resources if *rx* is zero (i.e., not used in the routing path of net

nx). In such a case, *Term2* would be zero and then (25) could be rewritten as *Term1* ≤ 2, which prevents vertex *v1* from being shared by multiple nets (i.e., the short circuit constraint for *v1*). In case *v2* is a terminal, it should be connected to just one segment. In this situation, if *rx* is one (that means *v2* belongs to net *nx*), *Term2* would be equal to 1 and (25) would be rewritten as *Term1* ≤ 1, which means *v1* cannot be used as a non-terminal in the routing path of other nets (i.e., *v1* reserved for *nx*). Similar to the non-terminal situation, if *rx* is zero, (25) might be seen as a typical short circuit constraint for *v1*.

Equations (25) and (26) are aimed at the wire width of 2 with just one of the vertices (i.e., *v1* in the example of Fig. 15) around the wider net (i.e., *nx* in the example of Fig. 15). These constraints can be readily expanded to cover any wire width larger than 2 by increasing the constraint range. If the wire width is equal to *wx*, the number of vertices to be constrained should be *wx*. Moreover, it is important to notice that segment set {0,1,2,3,4,5,6,7} described in Section 3.2.1 is divided into two segment sets {1,2,5,6} and {0,3,4,7} in (25) and (26). The reason is that we can achieve the horizontal path direction by using the first set {1,2,5,6} while selecting one of the segments from the second set {0,3,4,7} will lead to the vertical path direction.

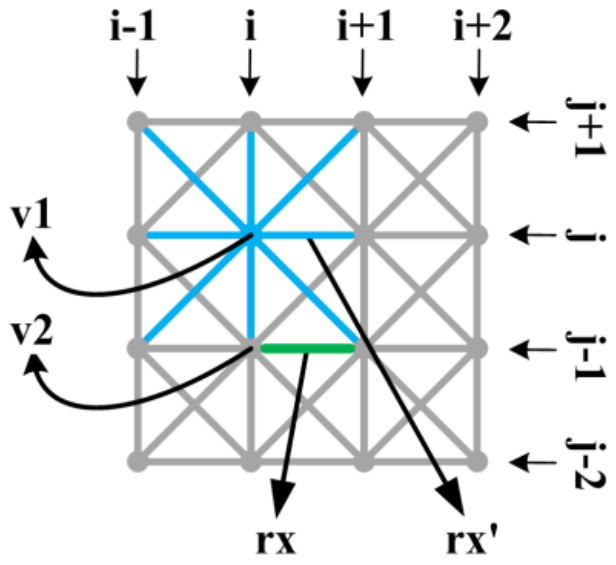


Fig. 15. An example of the wire width constraint.

In addition to the wire width control constraints, we have also proposed an ILP formulation for via widening constraints to address via voids. A similar vertex reservation idea utilized in our wire width formulation above is used for via widening, which means the vertices that need to make more space for via via_x belonging to net nx cannot be utilized as the routing resources of any other nets. Such via widening for via_x between layers k and $k+1$ can be formulated as follow:

$$\underbrace{\left(\sum_{n\in N}\left(\sum_{s\in S}(r_{v11,s,n})+via_{v11,n}\right)\right)}_{Term1}+\underbrace{via_x\wedge(\neg\ (r1_x))}_{Term2}\le 2, \tag{27}$$

$$v21=v_{i,j,k}\ , v11=v_{i+1,j,k}\ ,\ via_x\ links\ v21\ and\ v22\ ,$$

$$\left(\sum_{n\in N}\left(\sum_{s\in S}(r_{v12,s,n})+via_{v12,n}\right)\right)+via_x\wedge(\neg\ (r2_x))\le 2, \tag{28}$$

$$v22=v_{i,j,k+1}\ , v12=v_{i+1,j,k+1}\ ,\ via_x\ links\ v21\ and\ v22\ ,$$

where $\wedge$ and $\neg$ are the logical AND and logical NOT operations, respectively.

The example in Fig. 16 shows the details of the via widening constraint, by which more space is needed for via_x belonging to net nx. Here via_x is located between $v21$ and $v22$ on the planar location (i,j). To ensure an extra via on $(i+1,j)$ can be later added to net nx, two vertices

v11 and *v12* have to be reserved for *nx*. It is assumed that $r1_x$ ($r2_x$) is the segment that links *v21* (*v22*) and *v11* (*v12*) as shown in Fig. 16. There are two terms in (27), *Term1* and *Term2*. For *Term2*, if $r1_x$ has been selected as the routing path of net *nx* (i.e., $r1_x = 1$), it is not necessary to reserve vertex *v11* for extra via since it is already belonging to this net and in (27) *Term2* would be equal to zero. When *Term2* is zero, (27) can be rewritten as $Term1 \leq 2$, which is actually the short circuit constraint for vertex *v11*. However, if $r1_x$ has not been used as the routing path of net *nx* (i.e., $r1_x = 0$), *v11* should be reserved for net *nx* if there is a via on (*i*,*j*) (i.e., via_x is equal to one). In this case, *Term2* has to be one, which contributes to *v11* reservation for net *nx*. The same scenario can be expressed for (28), which is the formulation of *v12* reservation constraint. According to (27) and (28), *v11* and *v12* will not become the routing resources for the nets other than *nx* and the via between vertices *v11* and *v22* can be seen as via widening for via_x.

Equations (27) and (28) express via widening formulation for only one extra via, while those constraints can be easily expanded to support even wider vias (e.g., via arrays) by constraining more neighbor vertices. It is important to notice that if an extra via has to be located on non-neighbor vertex v_{nn}, *Term2* in (27) needs to be modified to $via_x \wedge (\underset{s \in S}{\wedge}(\neg\, r_{v_{nn},s,nx}))$, which means *Term2* is one only if via_x is equal to one and v_{nn} has not been selected as the routing path of net *nx*. Here $\underset{s \in S}{\wedge}(\neg\, r_{v_{nn},s,nx})$ is interpreted to be true if none of the segments connected to v_{nn} is selected as the routing path of net *nx*. A similar modification should be applied to (28) as well for non-neighbor vertices.

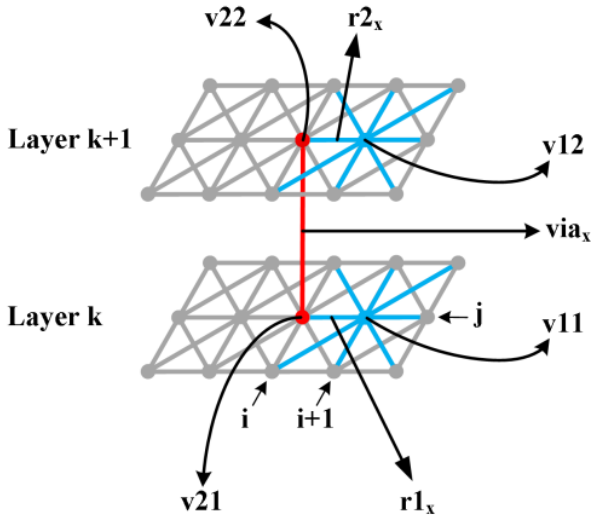


Fig. 16. An example of the via widening constraint.

By using the ILP width control formulation in (25)-(28), our proposed router is able to address electromigration, interconnect parasitic, and mismatch minimization (which would be discussed in the following section) as well as other analog/RF special constraints simultaneously. Moreover, the number of the constraints in the proposed ILP formulation remains the same compared to the case without using the wire width control constraints. The reason is that we have extended the short circuit constraints to develop wire width constraints instead of adding new constraints. In the example of Fig. 15, *v1* needs to have the short circuit constraint, which can be expressed by $Term1 \leq 2$ if using the terms from (25). Therefore, our proposed ILP routing method equipped with the wire width modeling is able to work efficiently since there is no extra constraint compared to the typical ILP modeling [31].

4.3. EM- And Parasitic-Aware Routing

4.3.1. Proposed Routing Flow

In our work interconnect (i.e., wire and via) width is treated as a parameter, which can not only address the EM issues but also improve the circuit performance in two ways, parasitic control and mismatch minimization. To include the EM consideration in our proposed router, a width control scheme has been developed to determine the width of each subnet based on its flowing current. Since the incurred parasitics change when interconnect widths vary for an electrical net, analog and RF circuits have to be routed with accurate interconnect widths. Thus, determining appropriate widths for interconnects is the first step that should be considered in the routing stage. The second important aspect of the parasitic-aware routing is to minimize mismatch effect of the nets that are supposed to be ideally matched. Our experiments show that there are always certain situations where sensitive matching nets need to be routed by applying different interconnect widths to satisfy the matching constraints, which are otherwise impossible to be realized by the regular analog routers. This means that the interconnect width is actually an effective way to generate a suitable solution for the ideally matching nets.

Fig. 17 shows our proposed EM- and parasitic-aware routing flowchart. The two major operations in this figure include interconnect width determination and mismatch minimization, both of which will be discussed in more detail below. The routing process is started by performing EM-aware initial routing, which utilizes specific interconnect widths that can satisfy minimum EM width as detailed in Sub-section 4.3.2. Then in the next stage called “Initial values for π model”, those initial interconnections derived from the EM-aware initial routing output are

used to estimate the parasitic resistance and capacitance (R&C) of each interconnect in the circuit.

In Section 4.3.3 the relationship between resistance/capacitance and wire width/length will be discussed (i.e., by (29) and (30)). This estimation would be used as interconnect parasitics in the schematic simulation for the next routing stage, which is called *sensitivity analysis*. Its purpose is to check how interconnect width may affect the circuit performance. Afterwards optimum width values would be determined for each net so that the best circuit performance will be attained in the present of interconnect parasitics. We integrate π model [73] to reflect the parasitic effect of physical wires into sensitivity analysis on schematic. One example of inserting π model into the interconnect *AB* is shown in Fig. 19. With the best values for interconnect widths, the mismatch problem (as discussed in Section 4.3.4) can further determine the interconnect width for the matching nets within the block of "EM-aware mismatch minimization" while still respecting EM rules as shown in Fig. 17. The last step is to call an ILP router to generate the routing results to realize the wire width and via width without EM failure in either wire traces or high current-density vias. The ILP modeling for wire and via width consideration has been explained in Section 4.2.1.

As discussed in Section 4.2.1, our proposed ILP router works on all the divided routing regions one after another. Since the interconnect width is specified before the final routing stage (as shown in Fig. 17), the default routing resolution of each routing region may not be sufficiently fine for all the interconnections. Therefore, the routing resolution of each routing region needs to be adaptively adjusted according to the finest routing resolution, which can be determined by the interconnection widths within that region. For instance, if in the initial routing stage the routing resolution of region *R1* is 1 unit and the finest interconnect width needs to be

1.5 unit, then the routing resolution of region *R1* should be doubled in the final routing stage in order to make sure that any wider interconnections will be routed accurately. Obviously the routing regions with the regularly-sized interconnections can be processed by the default basic routing resolution for reaching the best computational efficiency.

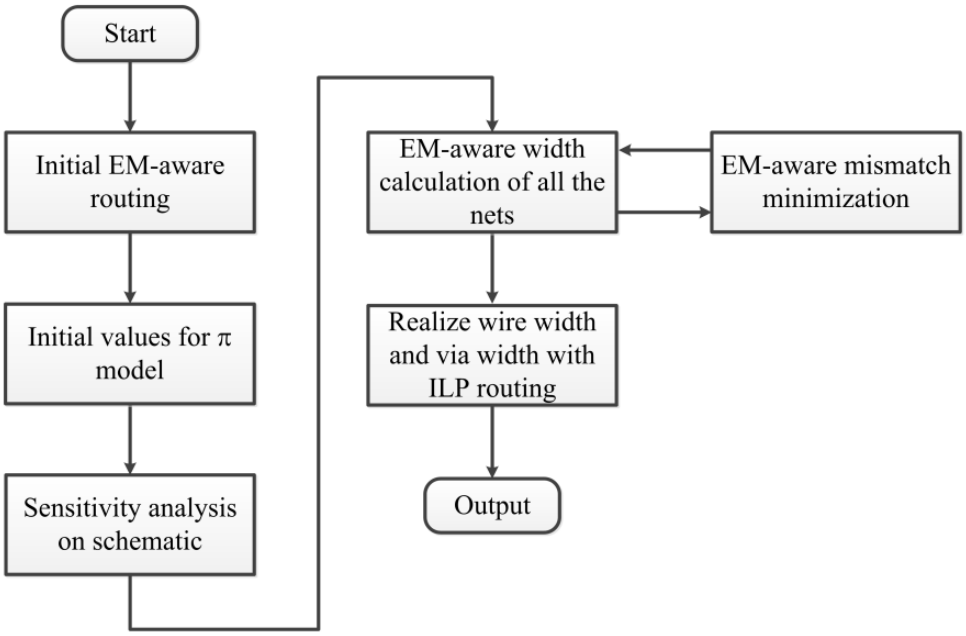


Fig. 17. Flowchart of the proposed EM- and parasitic-aware ILP routing.

Before going to the next sections, let us discuss a case study that shows to what extent parasitic R&C may affect the performance of analog/RF circuits. An interconnection in a single-ended LNA has been chosen for this case study as shown in Fig. 18. This interconnection, which is replaced by the π model in Fig. 18 to reflect the equivalent parasitic R&C from the layout, links the input inductance (i.e., *L1*) and the input transistor (i.e., *M1*). According to the Manhattan distances between the ends of the interconnection, we can estimate the maximum resistance and capacitance, which are 3ohm and 40fF in this study respectively. If we sweep the resistance from 0 to 3 ohm, the Noise Figure (NF) would be degraded from 1.78dB to 1.93dB,

while if we sweep the capacitance from 0 to 40fF, S11 would drop from -15.2dB to -8.88dB. This means that performance degradation due to the parasitic resistance and capacitance of the interconnection is inevitable if we fail to simultaneously consider both in the routing stage. In our proposed method, interconnect width as a variable would be determined before the ILP routing stage in order to contribute to the best circuit performance as discussed in Section 4.3.3.

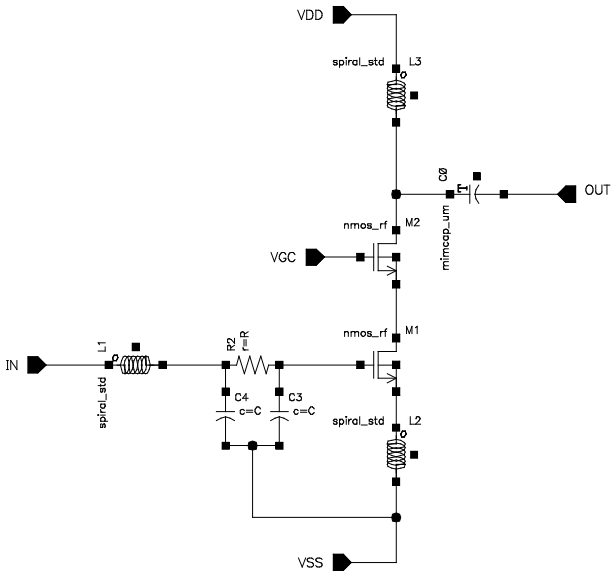


Fig. 18. Single-ended LNA with one input interconnect segment replaced by the π model as a case study.

4.3.2. Electromigration (EM)

The basics of EM and its corresponding equations have been explained in Section 2.1.1. In this work, we take into account the Blech length effect in our EM handling. For a given interconnect wire with a specific length, all the parameters in (1) and (2) except for the current density *j*, which is defined as the amount of flowing current per interconnect area, are constant. The interconnect area can be computed by $Area = ICT_{height} * ICT_{Width}$, where ICT_{height} is the

interconnect layer thickness, a constant for a specific metal layer. Thus, the only parameter, which can be used as variables for optimization, is wire width ICT_{Width}. The determination of ICT_{Width} should not only be based on the MTTF specification through (1), but also the Blech length effect through (2). Accordingly we can conclude that the only way to address the interconnect EM failure is to increase the wire width, which can result in less current density and in turn more MTTF as defined in (1) and (2). On the other hand, via void as the second EM issue can be addressed by using extra metal-to-metal vias (such as via arrays) to distribute current density among those vias. Therefore in this work, EM void problem can be addressed in our ILP formulation as discussed in Sub-section 4.2.1.

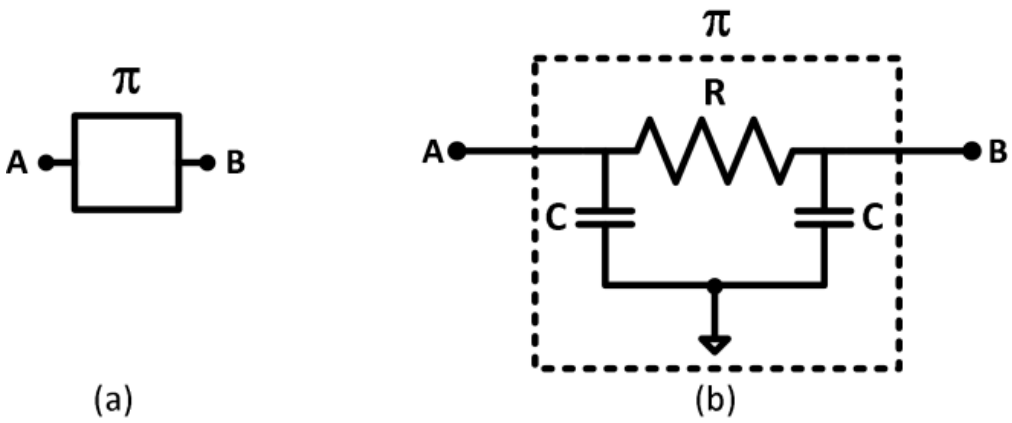


Fig. 19. (a) Two-terminal *AB* when its parasitics are considered by π model. (b) The schematic of the π model includes parasitic resistance (R) and capacitance (C) for *AB*.

4.3.3. Interconnect Parasitics

Due to the sensitivity to interconnect parasitics, analog and RF circuits may not be able to achieve the expected performance after the final routing. To address this issue, we include a sensitivity analysis operation in the routing flow [74], which is able to firstly generate a range for

resistance and capacitance of each interconnect and then determine an accurate interconnect width according to the R&C range. Interconnect R&C considerations can be done by replacing each interconnect with a π model block in the schematic. Our experiment shows that the sensitivity analysis would be more accurate when interconnect initial R&C values extracted from the existing layout are taken into account during the analysis. Therefore, we firstly route the nets with minimum allowed width values based on the EM formulation discussed in Section 4.3.2. Afterwards, the length and width of each subnet can be evaluated to derive initial R&C by using (29) and (30) [73] [75].

$$R_{Int} = R_{Sq} \frac{ICT_{Length}}{ICT_{Width}} \ , \tag{29}$$

$$C_{Int} = (ICT_{Length} * ICT_{Width}) \ C_{Ov} + 2 \ (ICT_{Length} + ICT_{Width}) \ C_{Fr} \ , \tag{30}$$

where R_{Int} and C_{Int} are the interconnect parasitic resistance and capacitance respectively, R_{Sq} is the sheet resistivity that can be extracted from the technology parameters for any interconnect layer, ICT_{Length} and ICT_{Width} are the length and width amount of the interconnect track respectively. And C_{Ov} is the unit overlapping capacitance and C_{Fr} is the unit fringe capacitance, both of which are constant for a specific technology.

In the sensitivity analysis stage, the resistance and capacitances in the π model are sweeping to identify their appropriate ranges. With such π model R&C ranges and interconnect length, the determination of the minimum interconnect width can be formulated as a nonlinear programming (NLP) minimization problem, with the following constraints:

$$R_{Min} < R_{Int} < R_{Max} \ , \tag{31}$$

$$C_{Min} < C_{Int} < C_{Max} \ , \tag{32}$$

$$ICT_{Width} > ICT_{Width-EM} \ , \tag{33}$$

where R_{Min} (C_{Min}) and R_{Max} (C_{Max}) are the minimum and maximum parasitic resistance (capacitance) for the R&C ranges, while R_{Int} and C_{Int} can be modeled by using (29) and (30) above, respectively. Constraint (33) ensures that the interconnect width will not be smaller than the minimum EM-compliant width (i.e., $ICT_{Width-EM}$).

The objective function of this NLP is to minimize the summation of all the interconnect areas, which can be expressed by (34).

Minimize

$$\left\langle \sum_{0 \le i < Q} (ICT^{\,i}_{Width} * ICT^{\,i}_{Length}) \right\rangle , \tag{34}$$

where Q is the number of interconnections and ICT^{i}_{Width} (ICT^{i}_{Length}) is the associated width (length) of interconnect track i. Here the interconnect length is the Manhattan distance between two ends of interconnect as per the initial routing stage and the interconnect width is the only variable parameter in the NLP optimization. Since the resistance and capacitance evaluated in (29) and (30) are layer dependent, it is worth noting that for each interconnect a list of widths corresponding to all the routing layers would be calculated.

4.3.4. Mismatch Problem

We have proposed a new methodology for finding the interconnect widths, which is able to minimize the mismatch impact of any sensitive nets. The core idea is to minimize the resistance and capacitance differences between the ideally matching nets whenever any typical matching

methods cannot reach an exact matching solution to certain cross-overlapping scenarios. To the best of our knowledge, none of the previous works (e.g., [39] or [41]) is able to consider such a mismatch issue, whereas our proposed methodology can detect such a situation and generate the minimum mismatch output. In this Sub-section, we will first describe the mismatch problem and then discuss our proposed method in more detail.

Since each net is comprised of multiple two-terminal subnets, two nets are defined as *matching nets* only if their corresponding two-terminal subnets are matched. The existing works from the literature describe the matching constraints as the same or correlated routing layer, wirelength, number of vias and bends, and sometimes direction for the matching nets. These constraints can ensure that those matching nets would be routed with the same or correlated physical specifications. In the following example we will show that there are always certain situations where such matching constraints cannot be satisfied in any case. Therefore, other solutions have to be sought for those sensitive nets.

As shown in Fig. 20, two nets *Net1* and *Net2* are ideally matched with reference to symmetry axis $X_{Net1,Net2}$ along with their terminals. *Net1* in gray contains terminals *A1*, *B1*, *C1* and *D1*, while *Net2* in red includes terminals *A2*, *B2*, *C2* and *D2*. $S1^1$ and $S1^2$ ($S2^1$ and $S2^2$) for *Net1* (*Net2*) are the Steiner points that are used to interconnect the terminals on the same net.

Let us look at the scenario exhibited in Fig. 20. Obviously in this example, *Net1* and *Net2* cannot be routed under the traditional matching constraints since these two nets have to cross each other for the basic commitment of routability. It means that if we route any of the matching two-terminals with the same wirelength as well as via and bend numbers, then their layers must not be the same, or if there is a solution to routing them on the same layer, then their wirelengths must not be the same. In this thesis we call this situation a *mismatch problem*, which may result

in different parasitic resistance and capacitance of some two-terminal subnets and in turn mismatch between *Net1* and *Net2*. Our experiments show that we should minimize such mismatch, which otherwise may deteriorate circuit performance.

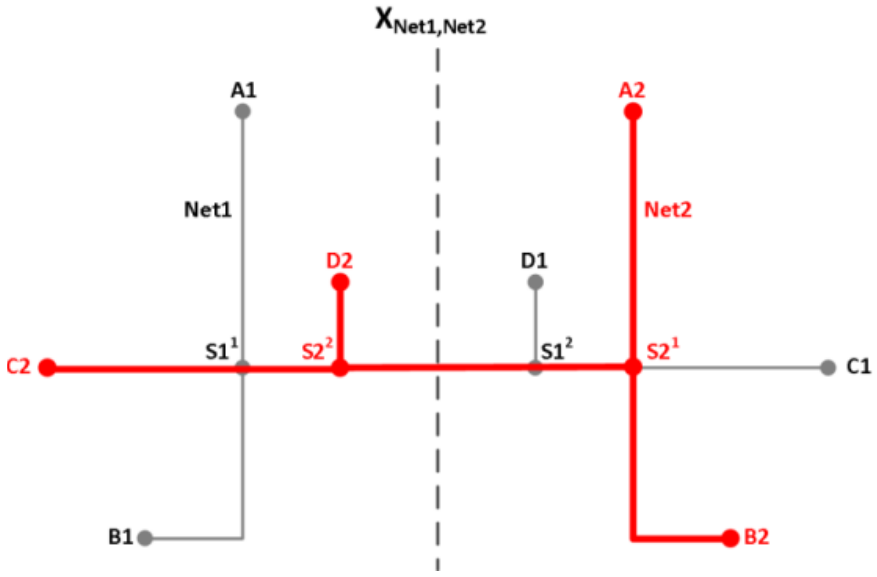


Fig. 20. Routing paths of two matching nets *Net1* and *Net2*.

For the mismatch problem discussed above, we have proposed a method to minimize its impact on the analog/RF circuits. For any matching nets, firstly Rectilinear Steiner Minimum Tree (RSMT) for each net will be generated. Then based on the symmetry axis (e.g., $X_{Net1,Net2}$ in Fig. 20), any two-terminal nets would be checked to see whether they have ends on both sides of the symmetry axis. If such an overlapping situation exists, the mismatch problem does appear due to the symmetric nature of these matching nets. In the example of Fig. 20, two-terminal subnets $S1^1$-$S1^2$ and $S2^1$-$S2^2$ are identified to be overlapping with each other. To resolve this problem, two overlapping subnets should be converted to a crossing two-terminal solution as illustrated below. Otherwise, the mismatch problem cannot be resolved.

To illustrate this, we will show the situations that may happen when routing these nets in a vertical symmetry-axis scenario, while the horizontal symmetry-axis scenario can be easily extended in a similar way. As detailed in Section 3.2.1, the routing paths in our routing model might be built up based on four types of orthogonal segments (i.e., $\rightarrow$, $\leftarrow$, $\uparrow$, and $\downarrow$). Assume the overlapping subnets (i.e., $S1^1$-$S1^2$ and $S2^1$-$S2^2$) are routed with orthogonal segments and the Steiner points $S1^1$, $S1^2$, $S2^1$, and $S2^2$ are utilized to ensure the minimum wirelength for the routing path. Because of the symmetric nature of *Net1* and *Net2*, for any horizontal segment $\leftarrow(\rightarrow)$ in *Net1*, a symmetric segment $\rightarrow(\leftarrow)$ should be selected on the same layer for the routing path of *Net2*. The same principle is also applied to the vertical segments.

Thus, two symmetric terminal-pairs should necessarily be routed within the routing area between two Steiner points $S1^2$ and $S2^2$, which in this thesis is called *overlapping area* where symmetry constraints cannot be simultaneously satisfied for ideally symmetric nets (e.g., *Net1* and *Net2* in Fig. 20) within N_{sym}. As a matter of fact, it is not possible to route such nets on the same layer due to the short circuit constraints. Even if we allow the matching nets inside the overlapping area to be routed on different layers, the vias on $S1^2$ and $S2^2$ might cause a short circuit between *Net1* and *Net2*. Moreover, the same via overlapping problem may take place for the other Steiner points, such as $S1^1$ and $S2^1$ in the example above. Below we first present a lemma and its proof for the mismatch minimization problem, and then our proposed method for resolving this problem will be discussed afterwards.

Lemma 1 (Mismatch problem): The matching nets with overlapping issues cannot be routed symmetrically if exclusively following the layer matching constraint in the context of the exact matching constraints for overlapping subnets.

Proof: The exact matching constraints include the matching of all the routing features (e.g., layer, wirelength, via number, bend number, routing direction, etc.) for the matching nets. Based on the discussion above, segments are selected symmetrically for the matching nets. There are at least two symmetric subnets that have to overlap with each other. To reduce the overlapping occurrence for making the problem solvable, they have to be routed by crossing segments. Since there is at least one point where those two matching nets are crossing each other, it is not possible to route them on the same layer due to the short circuit constraints if all the other factors of the exact matching constraints have to be closely followed. ■

To address this problem, we have proposed an algorithm, which is able to use diagonal segments (↖, ↗, ↙ and ↘) in our routing model to create crosses in the overlapping area. The key feature of the diagonal segment routing is to have the least intersection between two matching nets instead of multiple overlaps if utilizing orthogonal segments, which makes the matching problem less effective to be solved as described above. To generate a cross, Steiner points have to be rearranged around two selected terminals symmetrically with reference to the symmetry axis (e.g., $X_{Net1,Net2}$). In the example of Fig. 20, there are two possible solutions: either moving $S1^1$ and $S2^1$, or moving $S1^2$ and $S2^2$, symmetrically. To favor the least overlapping of subnets, movement of $S1^2$ and $S2^2$ should be discouraged since $S1^1$ and $S2^1$ can still form overlapping situations. Instead movement of $S1^1$ and $S2^1$ is favorable because of the least resultant overlaps as shown in Fig. 21. It is worth noting that all the two-terminal subnets except for the crossing ones (i.e., $S1^2$-$S1^3$ and $S2^2$-$S2^3$) in Fig. 21 can have the regular matching constraints. Although the crossing two-terminal subnets may have to be routed on different layers, they still can respect the other symmetry constraints (e.g., the same numbers of vias and bends as well as wirelength as

the symmetrical counterparts). We will discuss how to minimize the cross mismatch in the next Sub-section 4.3.5.

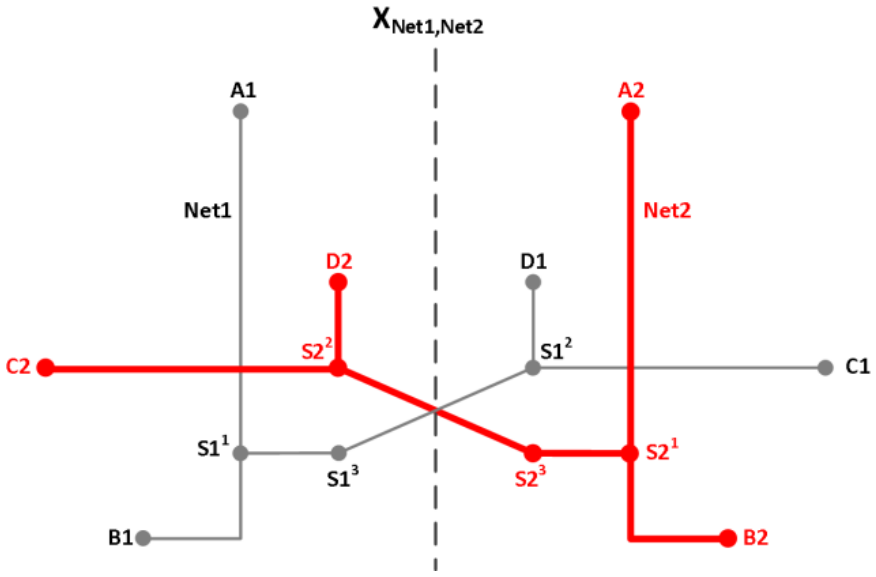


Fig. 21. Rearrangement of the Steiner points for the scenario depicted in Fig. 20.

4.3.5. Formulation of Mismatch Minimization

The conventional matching constraints may not be able to function when there is a physical overlap for the terminal coverage of the matching nets as discussed in Sub-section 4.3.4. We have formulated the mismatch minimization problem by using NLP. The main idea is to minimize the incurred parasitic resistance and capacitance of the crossing subnets. To represent the interconnect resistance and capacitance, we use (29) and (30) respectively.

If ICT_{Length} is known in (29) and (30), interconnect resistance (R_{int}) and capacitance (C_{int}) could be seen as functions of interconnect width ICT_{Width}. The constraint formulated in (35) guarantees that the minimum EM width rule is also satisfied,

$$ICT_{Width} > ICT_{Width-EM} \,. \tag{35}$$

The objective function of NLP for the mismatch minimization problem can be formulated as (36):

Minimize

$$\left\langle \gamma * \sum_{0 \le i < M} (R^i_{Int1} - R^i_{Int2})^2 + \eta * \sum_{0 \le i < M} (C^i_{Int1} - C^i_{Int2})^2 \right\rangle , \tag{36}$$

where R^i_{Int1} (C^i_{Int1}) and R^i_{Int2} (C^i_{Int2}) are the interconnect resistance (capacitance) evaluated by (29) ((30)) for the matching two-terminal subnet i. M is the number of two-terminal subnets associated with the matching nets. Factors γ and η are the used-defined coefficients. The designers can determine which parasitic element between resistance and capacitance is more effective as per the circuit performance sensitivity in the matching problem in order to choose proper values for γ and η. In the example above, interconnect $S1^1$-$S1^2$ should be matched with $S2^1$-$S2^2$, where four two-terminal subnets are involved. For instance, interconnect $S1^1$-$S1^2$ includes two two-terminal subnets, $S1^1$-$S1^3$ and $S1^3$-$S1^2$, where parasitic resistance (capacitance) R_{Int1_1} (C_{Int1_1}) and R_{Int1_2} (C_{Int1_2}) will be evaluated by using (29) ((30)). Interconnect length can be estimated as the Manhattan distance between two ends of each two-terminal subnet. After solving the NLP program formulated by (29)-(30) and (35)-(36), the width of each two-terminal subnet would be derived with a purpose of minimizing mismatch for the ideally matching nets.

Algorithm 4 describes the entire mismatch minimization flow for two matching nets *Net1* and *Net2*. First in Lines 1-5, all the two-terminal subnets would be checked to detect the overlapping ones. Then the vertices of graph *G(V,E)* that belong to the matching nets will be initialized based on RSMT of the matching nets. Each vertex *v* in *V* has an associated counter field that shows how many RSMT segments belonging to different nets exactly pass it. For example, since in Fig. 20 two nets (*Net1* and *Net2*) are sharing $S1^1$, the counter value for this vertex would be 2. On the

other side, vertex *B1* would have its counter field equal to 1. The next step of Algorithm 4 is to move one end (say, Sn^1) and its symmetric counterpart of the overlapping two-terminal subnets, while the other end (say, Sn^2) and its symmetric counterpart keep intact. Then based on the counter fields of the neighboring vertices for Sn^1, the one with the smallest count would be selected as a shifted candidate of Sn^1. It is clear that the shifted candidates are supposed to maintain their original connectivity among the electrical nets as before. Therefore, these candidates should be chosen from the vertices that are connected to the original nets within RSMT. For instance, $S1^1$ in Fig. 21 is a new location candidate for Steiner point $S1^1$ in Fig. 20.

Algorithm 4. Mismatch minimization

Input: Two-terminal subnets and symmetry axis of ideally matching nets *Net1* and *Net2*.

Outputs: Interconnect widths of the overlapping two-terminal subnets.
Overlapping area.

1	**For** (Any two-terminal subnet *TT1* belonging to *Net1* and *Net2*)
2	**If** (*TT1* crosses the symmetry axis)
3	Add *TT1* to *OvpTT*;
4	**End If**
5	**End For**
6	Initialize graph *G(V,E)* according to the RSMT;
7	Initialize *BestMov* as a sufficiently large positive integer;
8	**For** (Any *TT1* in *OvpTT*)
9	**While** (All routing area of *Net1*(*Net2*) is discovered)
10	Move one subnet terminal end of *TT1* (and its symmetric counterpart simultaneously) to its neighboring vertex *v* in *V* with the lowest counter field while maintaining the connectivity;
11	Find a path to *v* for *TT1* by using an A* algorithm and store the cost of the path into CT_{mov};
12	**If** (CT_{mov} < *BestMov*)
13	*BestMov* = CT_{mov};
14	Record *v* as the corresponding vertex for *BestMov*;
15	**End If**
16	**End While**
17	**End For**
18	Identify the overlapping area (*OA*), which is shared by two bounding boxes of the overlapping two-terminal subnets;
19	Formulate NLP based on the configuration stored in *BestMov*;
20	Resolve the formulated NLP problem and store the results of the overlapping two-terminal subnet widths into W_{ovp};
21	**Return** W_{ovp} and *OA*;

In the next step, the path from the other fixed end Sn^2 to the newly shifted location of Sn^1 is searched according to the A* algorithm. The cost of A* for a path from vertex *v1* to *v2* is computed as follows:

$$Cost(v2) = Cost(v1) + \psi * WireLength(v1, v2) + \zeta * Counter(v2), \quad (37)$$

where *Cost(v1)* is the A* cost of vertex *v1*, *WireLength(v1,v2)* is the wirelength from *v1* to *v2*, *Counter(v2)* returns the counter field value for vertex *v2*, ψ and ζ are the user-defined weighting factors. Therefore, the overlapping cost is reflected by the vertex counter field in this A* algorithm. Lines 9-16 of Algorithm 4 describe the process of finding the best vertex candidates for the subnet terminal shifting operation. Any possible vertex within the bounding boxes of *Net1* (*Net2*) will be discovered in Line 9 and the shifting operation is done in Line 10. Afterwards the A* algorithm identifies the minimum cost path in Line 11 and the solution with a lower cost than the previous ones would be recorded as shown in Lines 12-15.

Following the previous example depicted in Fig. 21, in Fig. 22 we include the graph vertices in order to demonstrate how the proposed A* algorithm can find the min-cost path between the two ends of the overlapping two-terminal $S1^1$-$S1^2$. The algorithm starts from $S1^2$ to look for the lowest-cost vertex based on (37). Obviously $V1^1$ gets outstanding and thus is selected. This is followed by $V1^2$ until the path ends at $S1^1$. The cost of this path would be the final cost for $S1^1$, which is stored in CT_{mov} as shown in Line 11. The overlapping area, within which certain relaxed constraints from the exact matching ones are applied as discussed in Sub-section 4.3.4, is extracted in Line 18 by using the shared bounding box of the overlapping two-terminal subnets. For instance, the overlapping area of $S1^1$-$S1^2$ and $S2^1$-$S2^2$ in Fig. 21 is the region surrounded by four corners $S1^3$, $S2^2$, $S1^2$ and $S2^3$. In Fig. 23, the bounding boxes of $S1^1$-$S1^2$ and $S2^1$-$S2^2$ are shown in dashed gray color with different stripe patterns, while their shared region is colored in black, which is actually the overlapping area. Finally the widths W_{ovp} of the overlapping two-terminal subnets would be derived by solving the NLP problem.

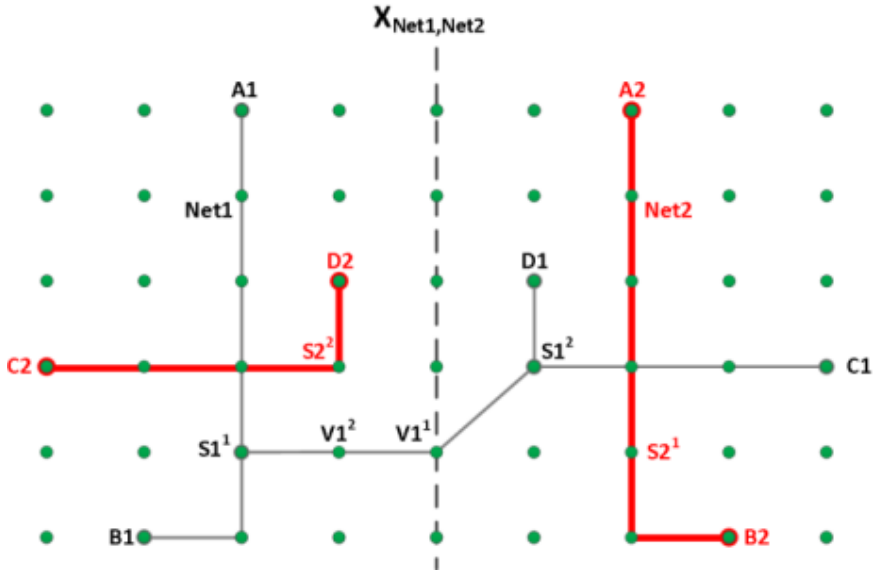


Fig. 22. A* algorithm path finding between $S1^1$ and $S1^2$.

Above we have discussed how the mismatch minimization would be resolved when diagonal segments are allowed in the routing path. In case only the orthogonal segments are desired in the analog/RF routing (e.g., for the highly advanced technologies where any diagonal segments may be strongly discouraged), our proposed method is still credible without losing the generality. Here we will briefly describe how the orthogonal routing can be also supported for the mismatch minimization. In the orthogonal routing, the general concept of Algorithm 4 remains the same if the overlapping area is modified to a bigger one. Here the combination of the bounding boxes belonging to the overlapping two-terminals would be selected as the overlapping area for the orthogonal routing, rather than the shared bounding boxes as discussed in Line 18 of Algorithm 4 for the diagonal routing. The overlapping area for the example above is shown in Fig. 24 when the electrical nets have to be routed only in the orthogonal directions.

In this work, we have proposed a scheme that minimizes mismatch by removing the layer matching constraint within the overlapping area, which cannot be maintained for the best matching effect anyway. Besides effectively addressing this problem, our proposed NLP

formulation could be adapted for other solutions, such as removing exact wirelength matching constraint by utilizing different interconnect widths to compensate wirelength mismatch.

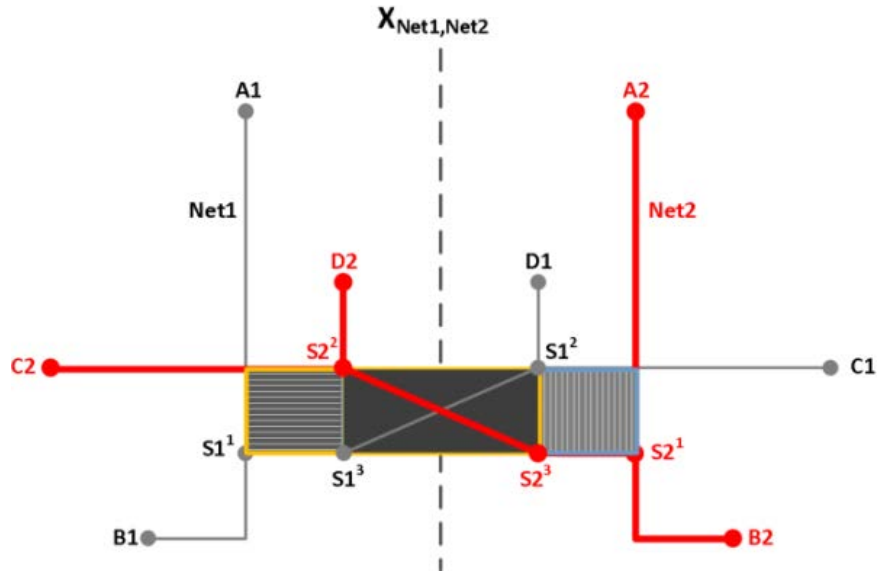


Fig. 23. Overlapping area (*OA*) of the diagonal routing.

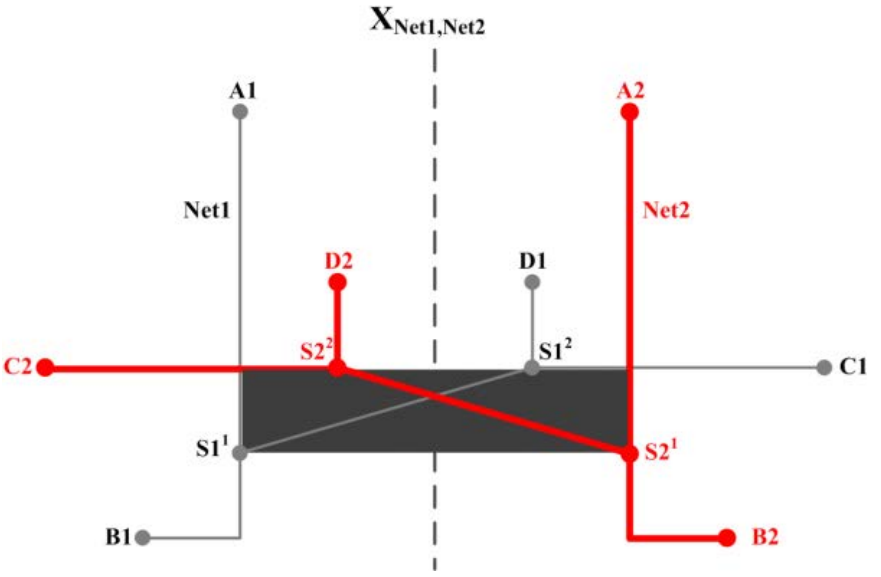


Fig. 24. Overlapping area (*OA*) of the orthogonal routing.

The symmetry constraint has been generally discussed in Sub-section 3.2.4 without considering mismatch minimization problem. In order to make ILP capable of handling the specific constraints for mismatch minimization, the symmetry constraint explained in (14) should be replaced by (38)-(39) for the vertical symmetry axis scenario, while the horizontal symmetry axis situation can be formulated in a similar way,

$$r_{v1,s1,n1} - r_{v2,s2,n2} = 0, \;\; s1 \in \{0,1,2,3\}, \;\; s2 = (8 - s1)\%\ 8 \;, \tag{38}$$

where

$$r_{v1,s,n1}, r_{v2,s,n2} \notin OA, \;\; (n1, n2) \in N_{sym} \;,$$
$$v1 = v_{i_1,j,k}, \;\; v2 = v_{i_2,j,k} \;, \;\; i_2 = (2 * X_{n1,n2} - i_1) \;,$$

where overlapping area set *OA* is the set of the vertices in *R*, which are located inside the overlapping area. One example of the overlapping area is shown in Fig. 23. On the other side, in the overlapping area the symmetric nets have to be routed with the regular symmetry constraint minus the layer matching constraint, which can be expressed by (39).

$$\sum_{k1 \in H} r_{v1,s1,n1} - \sum_{k2 \in H} r_{v2,s2,n2} = 0, \;\; s1 \in \{0,1,2,3\}, \;\; s2 = (8 - s1)\%\ 8 \;, \tag{39}$$

where

$$r_{v1,s,n1}, r_{v2,s,n2} \in OA, \;\; (n1, n2) \in N_{sym} \;,$$
$$v1 = v_{i_1,j,k1}, \;\; v2 = v_{i_2,j,k2} \;, \;\; i_2 = (2 * X_{n1,n2} - i_1).$$

4.4. Experimental Results

We implemented our proposed routing algorithm in C++ and the experiments were run on an Intel X86 1.2GHz Linux workstation that has 64GB of memory. We utilized IPOPT [76] for solving the NLP problems and Mosek 7 [71] for solving the ILP problems. Our experimental test

circuits include one analog dynamic comparator in a CMOS 45nm technology, one double-ended low-noise amplifier (LNA) in a CMOS 90nm technology, and one two-stage operational amplifier (OPAMP) in a CMOS 65nm technology. The schematics of the comparator, LNA and OPAMP are shown in Fig. 25, Fig. 26 and Fig. 27, respectively. To be fair in the comparison, the same manual placement input for each test circuit was provided to alternative routing methods in our experiments below.

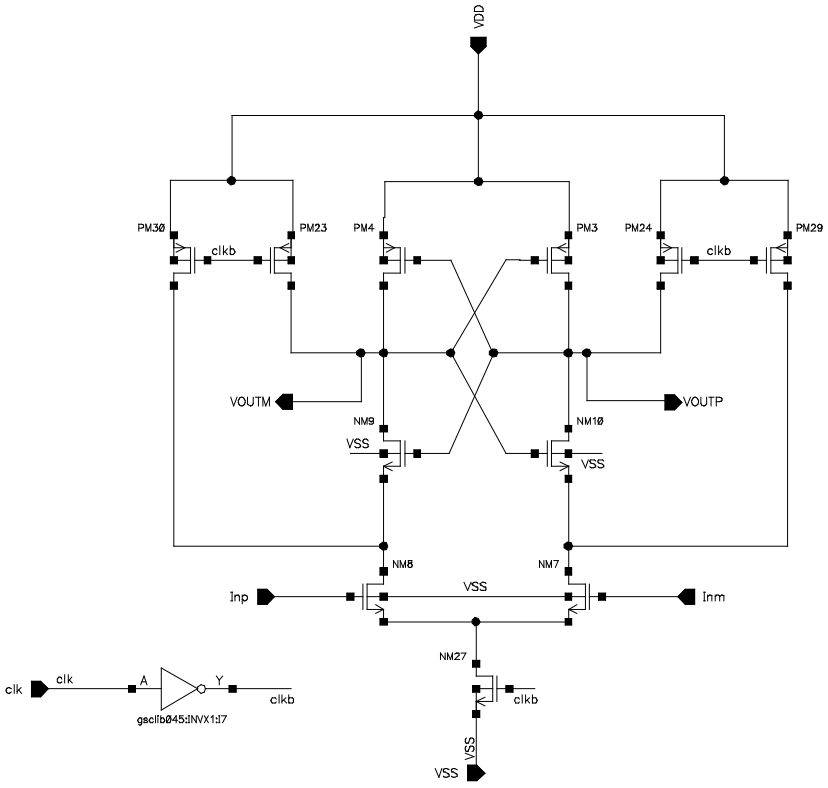


Fig. 25. Schematic of the analog dynamic comparator.

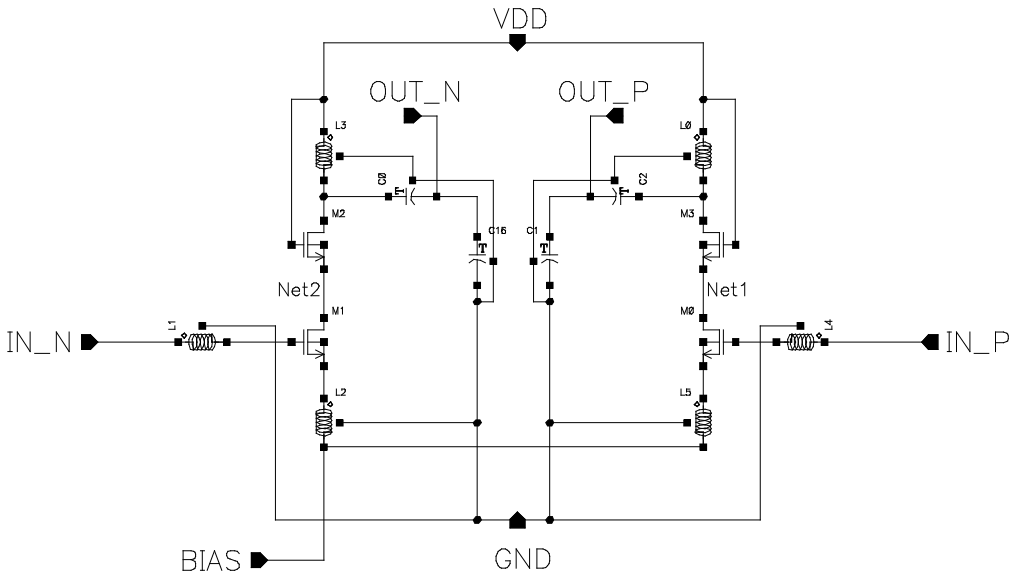


Fig. 26. Schematic of the double-ended LNA.

We used Cadence Virtuoso platform to extract the maximum current flow into each interconnect path. Then by using the EM formulation described in Section 4.3.2, we evaluated the EM-aware interconnect widths through both (1) and (2), where each interconnect length was derived from the Manhattan distance in the plain placement. Then based on the flowchart shown in Fig. 17, we ran an EM-aware initial routing to derive the interconnect widths without considering parasitic effects but only satisfying EM constraints. Afterwards as discussed in Section 4.3.3, an EM-aware NLP problem is formed to address the parasitic constraints for satisfying the given specifications. Moreover, the necessity of mismatch minimization was investigated for the matching occurrences in the given placement and terminal context for the circuits. In our experiments, the comparator and LNA circuits include cross-overlapping scenarios of ideally matching nets that needs to be handled by our proposed mismatch minimization scheme, whereas the OPAMP circuit only exhibits a need of interconnect width control for single nets without worrying about any mismatch problems.

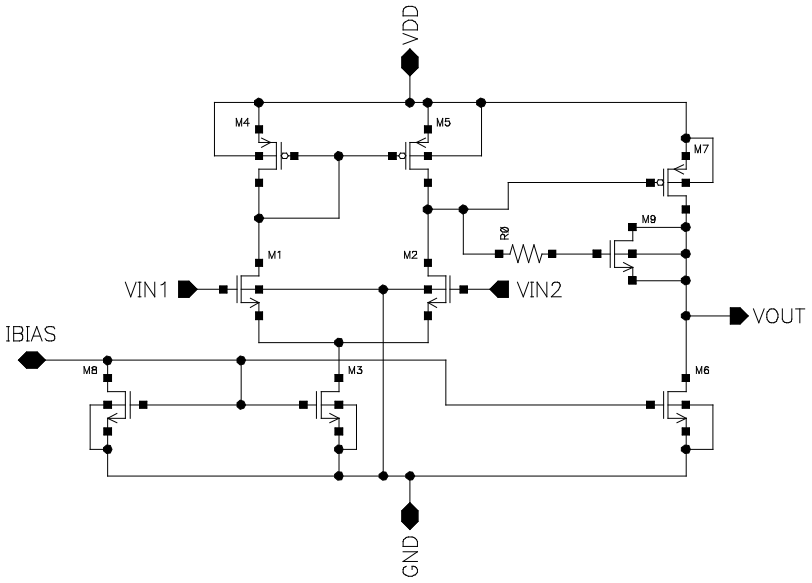


Fig. 27. Schematic of the two-stage OPAMP.

Circuit specifications ("*Spec.*" in short as column header), schematic pre-layout simulation performance ("*Sch.*" in short as column header), and post-layout simulation performance of different methods along with runtime and MTTF are listed in Table 5. Besides our proposed parasitic-aware routing method, we have also implemented a state-of-the-art analog routing scheme [31] (entitled as "*Regular-Routing*" as the column header in Table 5), which considers a variety of analog constraints but without accurate EM/parasitic control capability. By using this scheme, all the interconnects had to apply the same width due to the limitation of the ILP modeling. Moreover, we have also implemented another popular analog router [45] entitled as "*EM-Aware-Routing*" in Table 5, which can reduce the EM failure by widening the interconnect widths of the high current-density nets but missing the consideration of the Blech length effect. In addition, we have included the wire/via width control scheme but with neither NLP mismatch minimization nor our proposed ILP formulation for symmetric nets (as described in (38) and (39)) in another routing approach, which is entitled "*Width-Control-Routing*" in Table 5. Our proposed method is marked as "EM- and Parastic-Aware-Routing" [77] as column header in Table 5.

The post-layout simulation results are listed in Table 5. As regards the comparator circuit, *resolution* is defined as the minimum difference between two input voltages that the comparator is able to distinguish. For the Regular-Routing scheme, the minimum interconnect width from the technology design rules was applied. The resultant resolutions were equal to 195uV, 198uV and 182uV for the Regular-Routing, EM-Aware-Routing and Width-Control-Routing methods respectively, which failed to pass the specification. In contrast, when the circuit was routed by using our proposed mismatch minimization method, the resolution decreased to 35uV, which is about 5.6, 5.7 and 5.2 times better than the Regular-Routing, EM-Aware-Routing and Width-Control-Routing performances, respectively. The rationale behind these empirical findings is provided below. In the comparator charging stage, the capacitances associated with both output paths would be charged to the source voltage. Once the circuit turns into the discharging stage, both output capacitances would begin to discharge simultaneously. The better matching condition the both output paths can sustain, the higher resolution the comparator would achieve. Since there is an inherent overlap between these two paths, there would exist certain mismatch for sure, which can only be addressed by our proposed mismatch minimization scheme as discussed in Sub-section 4.3.4.

As for the LNA circuit, it has two groups of matching transistors that are desired to be in a common centroid structure to become immune to cross-chip gradient effects. Fig. 28 shows this structure, where transistors *M0* and *M1* are the first matching group while *M2* and *M3* are the second matching one with reference to the matching nets *Net1* and *Net2*. In Fig. 28, we use blocks "*M0.1*" and "*M0.2*" to represent the half layout modules of transistor *M0* inside the common centroid structure. The same naming convention is applied to the other blocks and transistors in this figure. Since there always exists certain overlap between these two nets, the

mismatch problem discussed in Sub-section 4.3.4 appears and needs to be resolved by our proposed mismatch minimization scheme. To be comparable to the example of Fig. 20, we utilize the same naming convention for the nets and Steiner points.

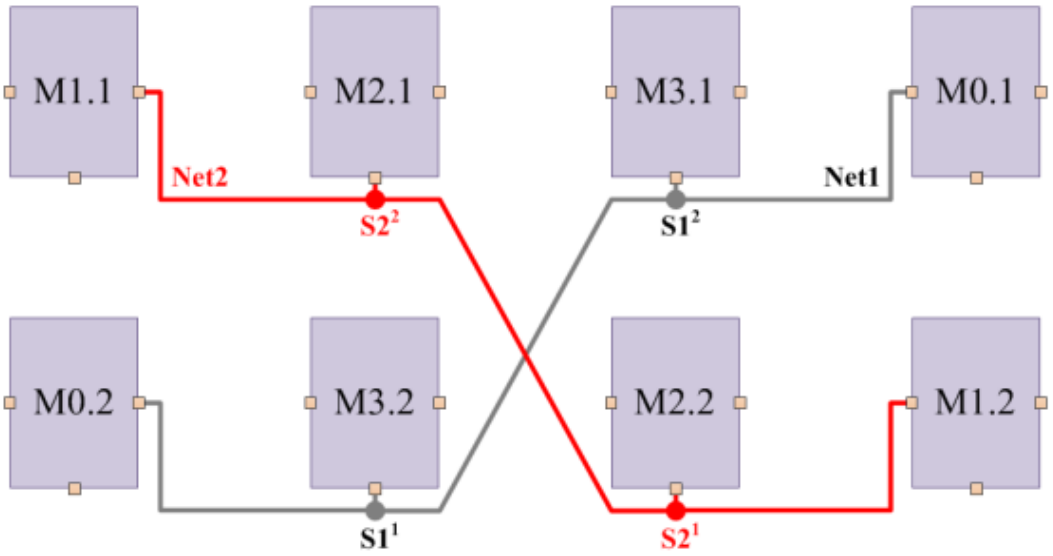


Fig. 28. Matching nets within the common centroid structure of LNA.

According to the experimental results in Table 5, since the interconnect parasitic effects were not holistically considered in the conventional Regular-Routing (EM-Aware-Routing) scheme, Noise Figure and S22 could only reach 2.4dB (2.3dB), and -6.05dB (-5.78dB) respectively, which failed to meet the specifications of the LNA circuit. Moreover, their Gain amounts are only 16.31dB for the Regular-Routing and 16.34dB for the EM-Aware-Routing scheme, which are 1.18dB and 1.15dB less compared to our proposed router. In contrast, our proposed interconnect parasitic-aware method could accurately set the interconnect width for each net, which resulted in satisfactory performance for the LNA well better than the given specifications.

Due to the interconnect parasitics in the OPAMP circuit as detected by sensitivity analysis, our proposed method was able to pass all the specifications while the Regular-Routing and EM-Aware-Routing schemes could only achieve the DC Gain of 59.8dB and 59.93dB respectively, which are less than the given specification. Moreover, we have included MTTF as a figure-of-merit to compare the failure effect of EM on interconnects. As shown in Table 5, the LNA and OPAMP circuits could work without failure for only 1.3 and 2.8 years respectively when routed by the Regular-Routing scheme, while our proposed method could improve to 10 years. On the other hand, due to the lack of the Blech length effect consideration in the EM-Aware-Routing scheme [45], this router was not able to fully satisfy the EM specifications for the LNA and OPAMP circuits (with 2.5 and 8.3 years respectively), whereas our proposed method could meet the specifications by achieving over 10 years of MTTF without fail. In addition, the EM-Aware-Routing scheme is obviously inferior to our proposed routing methodology in terms of the other circuit performances due to lack of interconnect parasitic and mismatch control. Moreover, as can be seen in Table 5, the runtime was increased by 6.5% (-5.7%), 17.9% (-8.2%) and 13.3% (21.4%) when the interconnect parasitic-aware and mismatch minimization scheme were used in our proposed router for the comparator, LNA and OPAMP circuits respectively compared to the Regular-Routing (EM-Aware-Routing) scheme.

Table 5. Pre-Layout and Post-Layout Simulation Performances of the Comparator, LNA, and OPAMP Circuits

Circuits		Spec.	Sch.	Regular-Routing [31]	EM-Aware-Routing [45]	Width-Control-Routing	EM- and Parasitic-Aware- Routing
Comparator	*Runtime (seconds)*	-	-	62	70	63	66
	MTTF (years)	10	-	10	10	10	10
	Resolution (uV)	50	18	195	198	182	35
	Falling edge delay (ps)	30	14.18	22.66	22.58	22.58	22.51
	Rising edge delay (ps)	40	19.99	37.89	37.9	37.97	37.91
	Positive falling-edge overshoot (mV)	50	38.25	21.28	20.76	20.69	19.79
	Negative falling-edge overshoot (mV)	100	83.58	38.47	38.68	38.43	38.28
LNA	*Runtime (seconds)*	-	-	234	255	270	276
	MTTF (years)	10	-	1.3	2.5	10	10
	Noise Figure (dB)	2.1	1.88	2.4	2.3	1.97	1.96
	Gain (dB)	16	18.24	16.31	16.34	16.91	17.49
	S11 (dB)	-10	-15.34	-6.05	-5.78	-8.64	-10.14
	S22 (dB)	-10	-15.23	-13.22	-13.25	-15.39	-11.95
	IIP3 (dB)	-6.5	-6.49	-5.28	-5.3	-4.95	-1.97
OPAMP	*Runtime (seconds)*	-	-	15	14	17	17
	MTTF (years)	10	-	2.8	8.3	10	10
	DC Gain (dB)	60	60.8	59.8	59.93	60.54	60.54
	Gain Margin (dB)	-10	-22.86	-12.54	-13.5	-13.59	-13.59
	Phase Margin (degree)	60	65.75	62.94	63.11	63.88	63.88
	Unit Gain Band- width (MHz)	300	376.8	341.9	342.2	343.5	343.5

4.5. Summary

In this chapter we have proposed a methodology that includes our unique contributions compared to the previous works for the analog/RF routing. We have formulated the problem first with NLP for routing planning and then with ILP for routing implementation. The proposed router can well control the layout interconnect parasitics especially beneficial to the analog/RF

circuits. Moreover, the parasitic mismatch can be effectively minimized when certain subnets along with their physical terminals of the ideally matching nets overlap with each other, which imposes difficulty in realizing the exact matching constraints for the primitive analog/RF routers. In addition, the EM effect has been considered as a constraint in all the routing stages to avoid any failure in either wires or vias. Our experimental results have shown that the circuit performance is largely improved by using our proposed method thanks to the special handling of analog and RF parasitic-related constraints. In the next chapter, LDE and OPC as two emerging analog layout challenges will be first discussed and then our proposed corresponding methodologies will be explained in detail.

Chapter 5 LDE-Aware Analog Layout Migration with OPC-Inclusive Routing

5.1. Introduction

Layout dependent effects (LDEs) are strongly associated with the advanced CMOS technologies. They used to be insignificant in the micrometer or above technologies and started to demonstrate their existence in the sub-micrometer technologies, whereas they become prominent in the state-of-the-art nanometer technologies nowadays. They may ruin the high-performance analog integrated circuits if not being properly handled. A similar scenario happens to the manufacturing lithography process. Although the manufactured pattern images were close enough to the ideal device layout for the outdated micrometer or above technologies, resolution enhancement techniques (e.g., optical proximity correction (OPC)) have to be resorted to for the advanced nanometer technologies at present.

Analog layout migration is a design methodology for analog layouts, by which the designers can use a silicon-proven input layout as a template to efficiently produce an optimal solution suitable for modified specifications and/or upgraded technologies. The capability of tackling LDEs and OPC is especially beneficial to the analog layout migration tools, which typically convert layouts from old technologies to advanced ones. In the old technologies, LDEs and lithographic discrepancies are normally not critical or even not required, whereas they become mandatory for the advanced technologies. Therefore, the analog layout migration tools demand such a specialization in dealing with LDEs and OPC as one prerequisite for them to be accepted by the analog designers in the era of advanced technologies. Our motivation of this chapter is to

combine analog layout redesigning process with LDEs and OPC handling by taking advantage of the special features offered by the analog layouts. Thus, the final generated layout can not only meet the updated specifications in a new technology but also sustain high image fidelity on the wafer.

In this chapter, we propose a new LDE-aware analog layout migration process which is integrated with an OPC-inclusive ILP-based routing scheme. To the best of our knowledge, none of the previous work in the literature has managed both LDEs and OPC in their physical design practices. According to our experimental results, these issues have to be taken into account to not only achieve better circuit performance, but also respect the manufacturability constraints for analog circuits. In this regard, our unique contributions are listed below:

- LDEs have been fully considered in the layout migration process to improve the circuit performance by using a smart sensitivity analysis.
- An OPC-inclusive ILP analog router has been developed to route the electrical nets by simultaneously considering routability, OPC, and other analog constraints.
- An extra-space reservation scheme is proposed to refine the placement so as to balance the trade-off between routability and silicon area.

5.2. LDE-Aware Analog Layout Migration Process

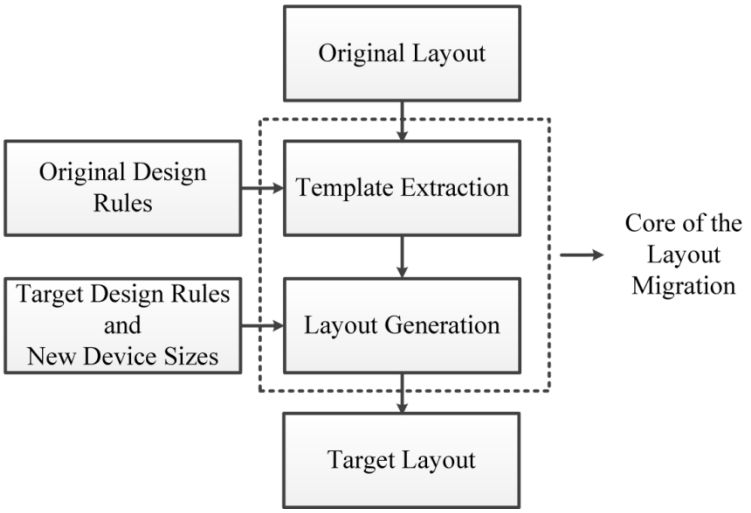
Rather than creating a layout from scratch, the analog layout migration process provides an efficient way of converting an existing analog layout design in an old technology to a new layout in the same or new technology for updated specifications. To reuse the designers' skills and experience reflected from the existing layout, its design template remains the same for the target

layout. The conventional analog layout migration flow is depicted in Fig. 29(a) [78]. The input of the entire analog layout migration process includes an existing layout, original technology design rules, and target technology design rules. The legacy knowledge from the existing layout is first extracted and stored as layout template. By integrating the target technology design rules and new device sizes, a target layout can be generated by following the extracted layout template.

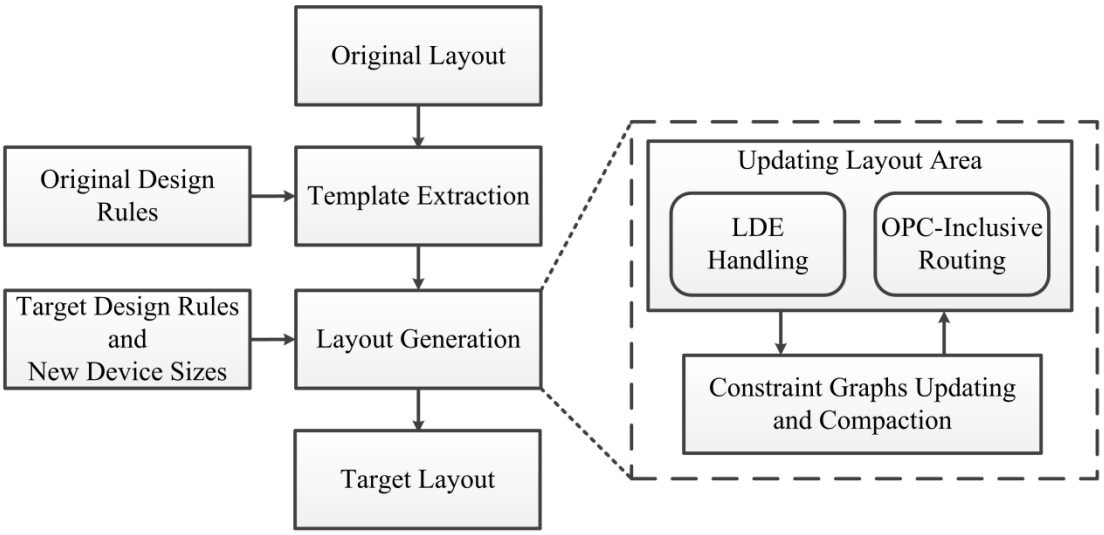
The flowchart of our proposed LDE-aware analog layout migration is illustrated in Fig. 29(b). The layout template, which is represented by constraint graphs, contains the information of device sizes, design rules, and analog constraints, such as matching and symmetry. Similar to the conventional layout migration process, the layout template is extracted from the existing layout according to the original technology design rules, which are afterwards replaced by the target technology design rules to form the target layout. However, different from the conventional layout migration process, the layout generation stage in our proposed flow has been largely reconstructed to include the LDE handling and OPC-inclusive routing for the advanced technologies. The LDE handling, which will be described in Section 5.2.2, is to ensure LDEs are considered in the layout migration. This is achieved by analyzing the effects of device LDE parameters on circuit performance and then adjusting those parameters for obtaining the optimal circuit performance. The OPC-inclusive routing, which will be discussed in Section 5.3, guarantees that there will be no lithographic hot spot in the layout neither inside devices nor among interconnect paths.

Due to the LDE and OPC considerations, the silicon area occupied by the related devices might have to increase. This is managed by the stage of *Updating Layout Area* as shown in Fig. 29 (b). Afterwards the resultant constraint graphs are updated and then compaction operations

are conducted by calling an efficient longest path algorithm. This will determine the location of all the geometric elements to form the target layout as the output.



(a)



(b)

Fig. 29. Flowchart of (a) the conventional analog layout migration, (b) our proposed LDE-aware OPC-inclusive analog layout migration.

5.2.1. Constraint Graph Representation

The constraint graphs are used to express the relative relationship between the geometric tiles, which represent devices or interconnect wires in a layout. For the sake of less complexity in the implementation, one horizontal and one vertical constraint graphs are deployed instead of a single complex two-dimension constraint graph [79]. The horizontal or vertical constraint graph is comprised of vertices and arcs. The vertices are used to represent tile edges or layout boundaries, while the arcs are aimed to mark the relative relationship between two vertices. There exist two types of arcs, *tile-arc* and *inter-arc*. The arc that links the edges of the same tile falls into the tile-arc type, while the arc that connects two different tiles or objects is categorized as the inter-arc type. The constraints of (or between) the tiles, such as design rules and analog special geometric requirements, are reflected as the weights of the tile-arcs (or inter-arcs).

Let us show the details of the constraint graphs by using the following example. Fig. 30 shows the horizontal constraint graph of a layout with four tiles. The vertical constraint graph, which is not shown in here, can be readily constructed in a similar way. In this horizontal constraint graph, there is one vertex for the left boundary and one vertex for the right boundary of the entire layout. Each tile has two vertices corresponding to its left and right edges respectively. And the arcs within the graph represent the constraints between each two vertices. For instance, arc A_B is a tile-arc that shows the minimum length of tile B while $A_{B\text{-}D}$ is an inter-arc whose weight represents the minimum distance between tiles B and D.

According to the target technology design rules and due specifications of analog circuits, the LDE-related geometric parameters are determined as discussed in Section 5.2.2. To meet the STI requirements, certain tile-arcs have to be updated in the constraint graphs. On the other hand, certain tile-arcs and inter-arcs need to be modified in the constraint graphs in order to satisfy the

WPE or oxide-to-oxide requirements. Once the constraint graphs are refurbished, the LDE-aware layout migration process can be partially completed by performing a longest path algorithm on the constraint graphs. The updated location of the tiles would be LDE-compliant in terms of the due specifications.

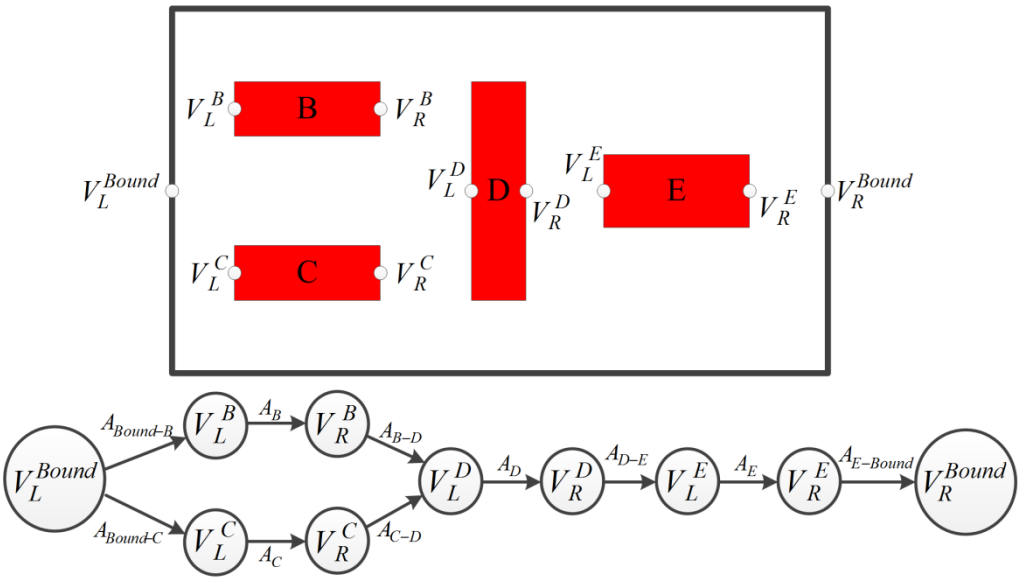


Fig. 30. The corresponding horizontal constraint graph of an example analog layout.

5.2.2. LDE Handling

As discussed above, LDEs are emerging as a result of three major sources, WPE, LOD and oxide-to-oxide. Due to strong nonlinearity of MOSFET and accuracy limitation of symbolic modeling, it is not possible to precisely predict how device LDE parameters can impact on the analog circuit performance. Therefore, we have proposed a special sensitivity analysis methodology that can tune the device LDE parameters to identify a set of suitable ones for optimal analog circuit performance. Here we briefly explain how the LDE parameters can be

tuned in our proposed sensitivity analysis method to present different scenarios for WPE, LOD and oxide-to-oxide considerations.

To address WPE, SCA, SCB and SCC as described in Sub-section 2.1.3 are targeted as the LDE parameters for tuning. In this regard, four edges of the rectangular NWELL blocks for each PMOS transistor are individually expanded by certain user-defined values, which can reflect various WPE scenarios leading to different SCA, SCB, and SCC. In detail, for the λ–based design rules of one specific technology node, the NWELL edges would be enlarged by λ, 2λ, 3λ, 4λ, and 5λ to make five incremental steps of sensitivity analysis for each PMOS transistor. The sensitivity analysis on LOD (as discussed in Sub-section 2.1.4) can be readily addressed by modifying the size of MOSFET active area to different user-defined values. Expanding such active area would increase SA and SB parameters as shown in Fig. 2, which results in less STI effect according to Equation (6). It is worth noting that the active area can be only enlarged in the MOSFET lateral direction to ensure the transistor size remains the same. In addition, the oxide-to-oxide sensitivity study can be managed by making more space between transistors with several user-defined increments during the sensitivity analysis.

As discussed above, the purpose of the sensitivity analysis here is to find out which set of device LDE parameters can bring about the most positive impact on the circuit performance. For instance, if two (or more) device LDE parameters have been identified with high impact on the circuit performance as per their individual sensitivity analysis, normally even more significant circuit performance is expected if those are considered at the same time. Moreover, according to our experiments, it is beneficial to also take into account the device LDE parameters with standalone moderate impact since they may lead to bigger impact on the circuit performance if being simultaneously considered with others due to the unpredictable nonlinearity among the

device parameters and circuit performance. Obviously the device LDE parameters with little or negative impact on the circuit performance can be ignored in the further sensitivity analysis. Since a simple brute-force traversal would lead to a huge number of LDE parameter combinations, such a straightforward sensitivity analysis with extensive simulations would be too expensive to afford. Therefore, we have developed a semi-dynamic-programming (semi-DP) algorithm to improve the efficiency of our proposed sensitivity analysis method.

In each step of the proposed semi-DP algorithm, the effect of any LDE parameter would be examined on the basis of the previously recorded performance improvement instead of starting from scratch. Thus, the evaluation efficiency can be significantly improved. Regarding the instances of the semi-DP, the smallest instances are the individual circuit devices, whose costs are calculated by doing simulation on them once a LDE parameter is changed. Then bigger instances can be constructed by merging the smaller instances. For example, if instance #1 is the LDE parameter tuning for device *D1* and instance #2 is the LDE parameter tuning for device *D2*, merging these two instances (to generate instance #3) means LDE parameter will be changing for both devices *D1* and *D2* simultaneously and then the cost will be evaluated by another simulation. Since the cost of instance #3 cannot be calculated directly from the costs of instance #1 and instance #2, here we have used the term of semi-DP instead of DP. However, we still utilize the reusing feature of the DP to reduce the number of simulations. More details of the proposed semi-DP will be explained in Algorithm 5.

As listed in Algorithm 5, the LDE parameter evaluation process takes an analog layout and matching-device-list as the input, and returns an LDE-aware device parameter combination as the output. In Line 1, we first initialize *DvGrpLst*, a list including all the non-matching devices and matching devices. Each group of the matching devices is treated as a single entry within

DvGrpLst. Then in Lines 2-5, a sensitivity analysis is performed on each device entry of *DvGrpLst*. In the following, we use the term of state to represent the instances of the semi-DP discussed above. If there is any cost improvement (i.e., a lower cost in our implementation), a state will be generated and recorded into *StateLst*, which is the list of semi-DP states containing the entries of sensitive devices along with their corresponding costs. For example, the multiple user-defined scales of NWELL boundary enlargement mentioned above are used for sensitivity analysis in Line 3 and then the cost would be evaluated at each scale accordingly as described in (40). It is worth noting that Line 4 would only get sensitive LDE parameters in terms of circuit performance into *StateLst*. The cost is defined by the following:

$$Cost = \delta * AI - \lambda * CPI \quad , \tag{40}$$

where *AI* and *CPI* stand for area increment and circuit performance improvement respectively, δ and λ are the user-defined coefficients. Since tuning the LDE parameters may be associated with layout area increment, we have considered both AI and CPI in the cost evaluation. In this regard, to obtain the minimum cost, a lower AI and higher CPI are desired. For instance, we may opt to enlarge the NWELL boundaries for PMOS transistors with several different values, which would result in larger AI, to decrease the WPE effect and in turn improve the circuit performances, which would accordingly increase CPI. We have deployed user-defined values for multiple NWELL boundary enlargement scales. For example, in 45nm CMOS technology these user-defined scales are 50nm, 100nm, 200nm, 300nm and 400nm in our experiments.

In Lines 7-15, the combination of different states within *StateLst* (such as *St1* and *St2*) will be considered. Here the combination of *St1* and *St2* means the LDE parameters will be tuned together for the devices in *St1* and *St2*, while the combined state is called (*St1, St2*). As described in Lines 8-12, the states are combined together and then the resultant state (i.e., (*St1, St2*)) will be

stored as a new state only if it brings about a cost that is not only lower, but also beyond the α-tolerance-range of the costs from the previous states. We can define cost value *CostB* in the α-tolerance-range of *CostA* when $CostA - \alpha < CostB < CostA + \alpha$. The idea behind this definition is to strengthen the semi-DP feature within Algorithm 5. It means that if a new state *StateB* with cost *CostB* has a cost almost the same as *CostA* (i.e., *CostB* is in the α-tolerance-range of *CostA*), this new state (*StateB*) should not be generated since such a cost level (i.e., α-tolerance-range of *CostA*) has been already reached by the previous states.

In Algorithm 5, the α value can affect the number of states and consequently the number of simulations. Therefore, we have deployed an adaptive method to update α according to the currently recorded states. At first α is estimated in Line 6 based on the cost improvement of the initial states within *StateLst*, while later it is adaptively updated in Line 15 once a specific state (i.e., *St1*) is combined by any other possible states. In our implementation, there are three user-defined updating options (i.e., the minimum, average, and maximum cost improvement values among the states) to select α. Finally, the state with the minimum cost within *StateLst* will be identified as shown in Line 17.

Algorithm 5. LDE-Aware Sensitivity Analysis

Input: An analog layout
List of matching devices

Outputs: LDE-compliant device parameter combination

1	Initialize *DvGrpLst* as the list of all the devices; /* Each group of the matching devices is treated as a single entry within *DvGrpLst* */
2	**For** (Any *DvG1* in *DvGrpLst*)
3	Perform sensitivity analysis on *DvG1* at multiple scales and evaluate the corresponding cost at each scale;
4	Store the cases with cost improvement as new entry states into *StateLst*;
5	**End For**
6	Estimate α based on the cost improvement in *StateLst*;
7	**For** (Any *St1* in *StateLst*)
8	**For** (Any *St2* not equal to *St1* in *StateLst* and not combined together before)
9	Evaluate the cost of the combined state (*St1, St2*) and record it as *NewCost*;
10	**If** (*NewCost* is lower and not in the α-tolerance-range of any cost of the previous states
11	in *StateLst*)
12	Add (*St1, St2*) as a new state to *StateLst* along with its cost;
13	**End If**
14	**End For**
15	Estimate α based on the cost improvement in *StateLst*;
16	**End For**
17	Find the minimum cost among the elements within *StateLst* and store its corresponding solution into *LDELayout*;
18	**Return** *LDELayout*;

The proposed algorithm above can efficiently examine and tune the related LDE parameters of sensitive devices in the analog circuits that lead to the minimum silicon consumption with the optimum electrical performance. This is done by using the concept of dynamic programming to quickly approach the best option without traversing all the possible combinations. Moreover, it is important to notice that Algorithm 5 can be easily extended to include other LDE sources, such

as Dual Stress Liner (DSL) [80] and Poly Space Effect (PSE) [59], in case certain circuits or advanced technologies show such a demand. After getting the position of each device from the resultant LDE-compliant layout migration, the OPC-inclusive routing, as discussed in Section 5.3, will be performed to derive the complete layout.

5.3. OPC-Inclusive Analog Routing

5.3.1. OPC-Inclusive Analog Routing Problem Formulation

Since in this chapter we will propose an OPC-inclusive routing scheme, the OPC-aware analog routing problem (explained in Section 2.2.1) can be redefined as "*finding the most suitable set of edges in graph G that can define paths among the terminals of the electrical nets besides satisfying routability and other analog constraints while no OPC hot-spots remain*".

5.3.2. OPC-Inclusive Routing Flowchart

In this section we will describe how OPC can be addressed during the process of analog routing. The flowchart of our proposed OPC-inclusive analog routing is shown in Fig. 31. The first step is to run OPC simulation on the initial LDE-compliant layout to remove the hot spots inside devices. In this chapter a device, which includes hot spots inside, is called *hot-spot device*. Those lithographic hot spots located inside the hot-spot devices are firstly examined and removed before the hot spots associated with the routing interconnects. This is because the hot spots inside are often caused by the limited space within the hot-spot devices, which cannot normally be removed unless additional space is granted. The regular analog layouts are typically much sparser compared to the digital ones, especially for the interconnect regions. However, this

is usually not true for the regions inside the devices, which are normally as dense as possible to save the silicon area if no performance degradation is incurred. Therefore, more space has to be sought to suppress these lithographic defects inside the hot-spot devices if any.

To understand how much extra space among the compact tiles inside the hot-spot devices is needed to suppress such defects, we have deployed a rule-based method [81]. In this regard, for one specific technology we first generate a set of rules that can define OPC hot-spot-free single tiles or tile compounds. This process is just a one-time effort so that the generated rules can be reused for the same technology once formed. By running OPC simulations on a large group of single tiles and tile compounds [62], we can extract the relationship between geometry size/distance and hot-spot-free enlargement threshold, which are stored into a look-up table. For instance, one rule has been established for two parallel tiles, whose individual widths and the distance in between are recorded in the loop-up table along with the width/distance enlargement solutions to avoid the open and short OPC hot spots.

For the hot spots inside the devices, the surrounding geometry around the hot spots will be first analyzed. Then the corresponding rules will be applied to the constraint graph (as discussed in Sub-section 5.2.1). For instance, if a hot spot is identified between two parallel tiles, the two-parallel-tile rule will be applied to modify the corresponding arc weights in the constraint graphs to new values according to the width/distance enlargement threshold stored in the rule-based look-up table. After applying them by executing a longest-path algorithm for the constraint graphs, there will be no hot spots left inside the devices.

After cleaning up the hot-spot devices, the hot spots in the routing interconnects need to be addressed in the next stages. Since the OPC-inclusive routing operation is applied onto the layout after the LDE handling as explained in Sub-section 5.2.2, “Constraint graph updating and

compaction" in both flowcharts as illustrated in Fig. 29 and Fig. 31 are the same. In the initial routing stage, we run our ILP-based analog router (as discussed in Chapters 3-4) to find a solution that satisfies both routability and analog constraints. Here we call the routing regions, which include lithographic hot spots, as hot-spot regions. It is more beneficial to consider the hot-spot regions all together in groups rather than separately since the OPC simulation results of various hot-spot regions might affect one another. In particular, some hot spots may appear between the neighboring regions, where grouping those regions may help improve the efficiency by invoking fewer OPC simulations. Thus, after finding all the hot-spot regions, a set of groups called *super regions* are generated for the next operations.

As shown in Fig. 31, for any super region (i.e., *SR1*), the electrical nets are routed while our proposed *space reservation* method (as described in Sub-section 5.3.3) is utilized to remove the lithographic hot spots. After routing all the super regions, another OPC simulation is needed to make sure the new routing solution does not generate any new hot spots. Otherwise, there will be another set of super regions in the layout and the same operations discussed above will have to be repeated. Since the electrical nets may be routed on higher metal layers due to the space reservation constraints (as discussed in Sub-section 5.3.3), our proposed OPC-inclusive router will always check whether there is any un-routed regions. The routing regions, where our router cannot find enough routing resources, would be reported as *unrouteable regions*. For them, more routing resources need to be granted by allocating more space among devices as shown in "Enlarging space among devices" block in Fig. 31. In this operation, the inter-arcs between the related tiles are marked with their weights added as per the estimation from the unroutable regions. Since such an operation actually affects the placement, this routing process should return to the beginning for another round of constraint graph updating and compaction to make

the routing resources available. The final operation is to call an *Aid Geometry Insertion* tool (e.g., Mentor-Graphics Calibre nmOPC [82] in our implementation), which can generate the OPC aid geometry patterns in the final layout. Since the extra space has been allocated in advance, we can ensure that there will exist no OPC hot spots in the migrated layout but the image fidelity can be significantly improved.

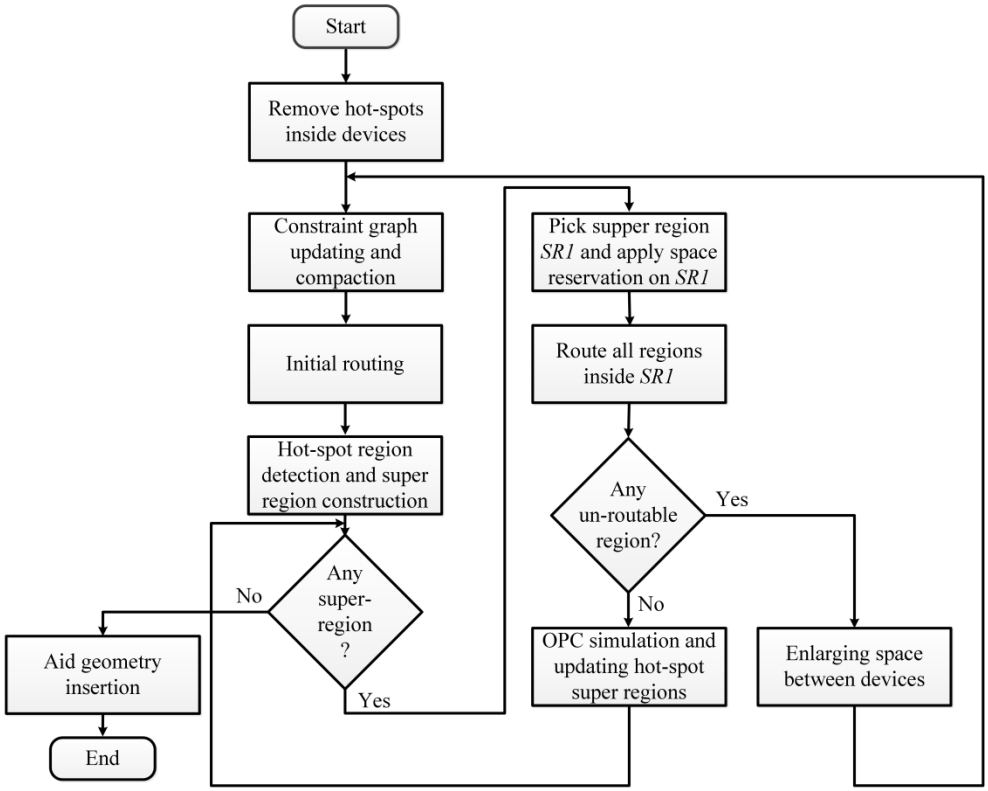


Fig. 31. OPC-inclusive analog routing flowchart.

5.3.3. Space Reservation

The idea of making extra space for OPC hot-spot removal is called *space reservation*. To address the closeness of the related tiles in terms of OPC hot spots emerging in the form of short or open circuit, we have developed such a space reservation scheme so that more space would be

provided to the electrical nets associated with OPC hot spots (called *hot-spot nets* accordingly throughout this chapter) with relation to the faulty tiles. The main idea is to route the electrical nets with extra space constraint to avoid such closeness.

For example, Fig. 32 shows two different situations of the tiles that are named (a) and (b) as well as their positions after applying the space reservation scheme as shown in (a') and (b'), respectively. For simplicity, we just assume that these tiles, as part of the routing paths, are routed with minimum distance between two electrical nets as per the technology design rules. In Fig. 32(a) two tiles are routed so closely that one hot spot is identified between them by OPC simulation. To remove this hot spot, we can utilize the space reservation scheme by selecting one of the tiles (e.g., Tile-1) and applying a certain extra space constraint. As shown in Fig. 32(a'), space reservation is applied on the right hand side of Tile-1, which is colored in yellow. Here the extra space on Tile-1 can be interpreted as the reserved extra space for routing Tile-1. Consequently, Tiles 1 and 2 cannot be routed closely as before due to this modified minimum distance design rule constraint, which will help eliminate the hot spot in between. In Fig. 32(b), the middle Tile-2 has been chosen for space reservation on its both sides to remove the hot spots among all these tiles. Therefore, the extra space on Tile-2 guarantees that both Tiles-1 and 3 have to be routed not closely to Tile-2 so that the hot spots will not remain consequently.

Moreover, the amount of the space reserved for the tiles is also determined by the rule-based method discussed in Sub-section 5.3.2. According to the specific geometric locations of the adjacent tiles, the built look-up table is utilized to identify a proper enlargement threshold value for space reservation. Since the tile routing solutions contribute to the final routing paths for each electrical net, our proposed space reservation scheme above is actually applied to all the electrical nets. In this thesis our ILP wire and via width control formulations discussed in Sub-

section 4.2.1 are used to reserve extra space for the hot-spot nets. Moreover, it is worth noting that a matching net should be selected for space reservation operation as well along with its counterpart net in order to maintain the matching constraints.

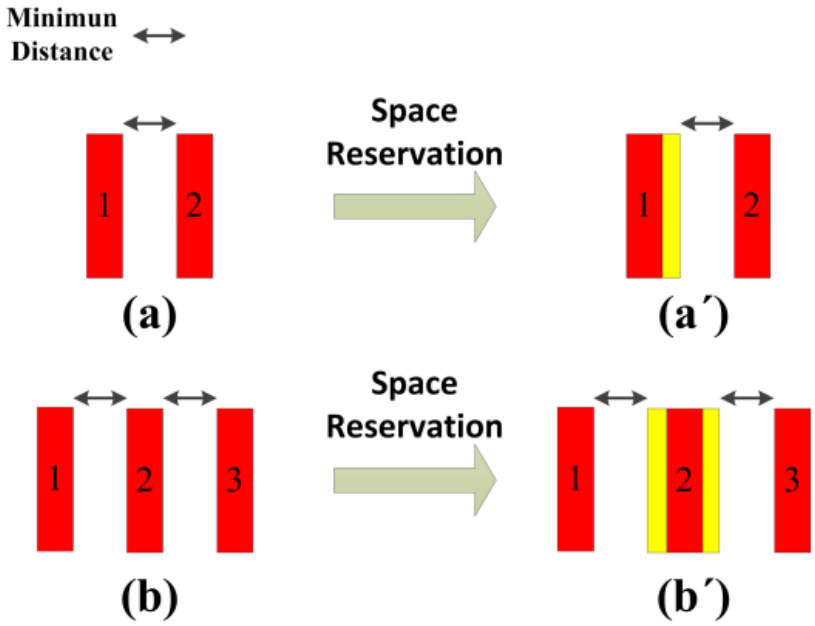


Fig. 32. Two situations of relative tile position in (a) and (b) along with their space reservation solutions in (a´) and (b´), respectively.

According to the discussion above, our proposed space reservation scheme would help remove the OPC hot spots at the cost of extra space for certain electrical nets. However, there are many options to choose which nets are less costly to be constrained for space reservation. Thus, it is important to determine the best set of nets for applying the space reservation as it may affect the final OPC-inclusive routing result. To this end, we have developed a selection algorithm that can find the best set of electrical nets for the space reservation in a super region. Algorithm 6

describes the net selection scheme for the space reservation, while the wirelength of the hot-spot nets is used as the cost parameter for net selection. In more detail, the algorithm gets the routing resources according to the initial routing results (as shown in Fig. 30) and a super region as input and returns the best set of electrical nets for the space reservation operation as output.

In Algorithm 6, the hot-spot nets within a specific super region are stored onto a list named *LstHSNets* as stated in Line 1. If one hot-spot net is selected as a space reservation candidate, it will be routed with wider width (i.e., by applying the wire and via width constraints as discussed in Sub-section 4.2.1) and thus its neighboring nets will have to be affected (i.e., being routed with larger distance from the hot-spot net). An array called *LstSRImptNets* in Line 2 will record all these affected nets for each hot-spot net, whereas the rest of the electrical nets are stored into *LstSRMissNets* as described in Lines 2-3. Each hot-spot net is attributed with a cost equal to its wirelength if all the hot spots are located just on one side of this net (i.e., the situation of Fig. 32 (a´)). If both sides of a hot-spot net are involved in the hot spots, its cost should be equal to double of its wirelength (i.e., the situation of Fig. 32 (b´)) as shown in Line 4. This is because constraining the shorter electrical nets would impose less complexity to the ILP modeling compared to the longer electrical nets. Moreover, smaller routing area inside the super regions would be affected if the shorter electrical nets are selected for the space reservation operation.

We have deployed a dynamic programming scheme in Lines 6-18 of Algorithm 6 to generate the least possible number of net combination so as to obtain the best set of nets for the space reservation operation. Now we explain how such a dynamic programming method works. The final solution to the space reservation operation would be saved into array *BstSRNets* that is firstly initialized to *LstHSNets* in Line 5. In Line 6 all the elements of *BstSRNets* are searched until no unchecked item is left. As listed in Line 7, item *i1* is selected as the first item of

combination. Then it will be combined with the other item (i.e., *i2*) to generate a new item called *i1i2*, if not being already existing with the same (or lower) cost on the list to maintain the dynamic programming feature according to Line 11. The reason for the combination is to build up a set of electrical nets that include all the hot-spot nets. It is obvious that the cost of the new item *i1i2* is equal to the summation of the costs of *i1* and *i2*. Afterwards *BstSRNets* will be updated to the electrical nets that are affected by both *i1* and *i2*, and meanwhile *LstSRMissNets* and *CostSRNets* would be updated as shown in Lines 12-14. Finally, the best set of the space reservation nets with the minimum cost is selected as shown in Line 19.

Algorithm 6. Net Selection for Space Reservation Operation

Input: List of routing resources for the super region
List of hot spots inside the super region.

Output: Best net selection set for the space reservation operation of the super region.

1	Determine the list of hot-spot nets for the current super region and save them into *LstHSNets*;
2	Initiate the corresponding *LstSRImptNets* as the list of the impacted nets that will be affected for each hot-spot net if the space reservation operation is applied;
3	*LstSRMissNets* = *LstHSNets* - *LstSRImptNets*;
4	Set the cost of each hot-spot net in *CostSRNets* to its wirelength if the hot-spots are located on one side of the hot-spot net, otherwise to wirelength*2;
5	Define *BstSRNets* as the best set of net candidates for the space reservation operation, initial it to *LstHSNets* and mark its items as unchecked;
6	**While** (There is an unchecked item in *BstSRNets*)
7	Pick up the first unchecked item *i1* in *BstSRNets*;
8	**For** (All the elements of the corresponding item within *LstSRMissNets* associated with item *i1*)
9	Pick up item *i2* and merge it with item *i1* to generate a new item *i1i2*; // As the list of the nets belonging to both items *i1* and *i2* to be selected for the space reservation
10	Set the cost of *i1i2* as the summation cost of *i1* and *i2*;
11	**If** (There is no heterogeneous item in *BstSRNet* with the same or lower cost)
12	Add *i1i2* to *BstSRNets* as a new item;
13	Add the missing nets of *i1i2* as a new corresponding item to *LstSRMissNets*;
14	Add the cost of *i1i2* as a new corresponding item into *CostSRNets*;
15	**End If**
16	**End For**
17	Mark *i1* as a checked item;
18	**End While**
19	Find the minimum-cost item within *CostSRNets* among the items with the corresponding empty item within *LstSRMissNets* and return its correspondence item within *BstSRNets*;

5.4. Experimental Results

We implemented our proposed LDE-aware analog layout migration with ILP-based OPC-inclusive routing algorithm in C++. The experiments were run on an Intel X86 1.2GHz Linux workstation that has 64GB of memory. We utilized Mosek 7 [71] for solving the ILP problems and Mentor-Graphics Calibre nmOPC [82] for OPC simulation. In our experiments we have included three test circuits, one analog dynamic comparator, one folded Cascode operational amplifier (OPAMP), and one two-stage OPAMP. All the circuits were retargeted from CMOS 0.18 um technology to CMOS 45 nm technology.

Circuit specification ("*spec*" in short as column header), post layout circuit performance for the layouts when LDE was not considered in the conventional layout migration by following [78] (entitled as "*Regular Migration*" in column header), and our proposed LDE-aware layout migration ("*LDE-Aware Migration*" in short as column header) are listed in Table 6. According to Algorithm 5 in Sub-section 5.2.2, there are three options for choosing α: minimum, average, and maximum for the purpose of cost calculation. In Table 6, we have included minimum and average values for comparison, which are entitled as "Min. α-range" and "Avg. α-range". In these two cases, Algorithm 5 was terminated when no further cost improvement was obtained. The maximum α-tolerance-range is not reported in the experimental results due to the page limit since it has similar results to the average α-tolerance-range for our test circuits. In addition, "*Sat. Spec.*" column in Table 6 is the resultant layout performance in the scenario where Algorithm 5 stopped running right away once the circuit specifications were satisfied.

Resolution has been defined as the minimum difference between input voltages that could be recognized by the comparator. According to Table 6, due to the lack of LDE consideration, comparator layout could not pass the specification for resolution with 197uV, whereas the LDE-

aware layout migration could meet the specification by improving the resolution to 170uV. In addition, for both Cascode and two-stage OPAMPs, DC gain could reach no more than 48.04dB and 49.15dB respectively, both lower than the specifications, when LDEs were not taken into account. On the other hand, our proposed LDE-aware layout migration scheme contributed to more than 50dB DC gain for both these circuits.

As can be seen, the numbers of the required simulations in the minimum α-tolerance-range option are 115, 2581 and 340 for the comparator, Cascode OPAMP, and two-stage OPAMP, respectively. Since the highest resolution is used in this option, slightly better circuit performance was achieved at the cost of 2-10 times more simulations in comparison with the other options (i.e., Avg. α-range and Sat. Spec.). Moreover, the minimum α-tolerance-range option increased the layout area by 0.93%, 1.81% and 0.38% for the comparator, Cascode OPAMP, and two-stage OPAMP, respectively. In contrast, the average α-tolerance-range option could derive much less layout area increment compared to the minimum α-tolerance-range option with only 62, 275 and 90 simulations required for the three test circuits. As regards Sat. Spec., we can see that, by running the least number of simulations, the specifications were satisfied with the least layout area increment (less than 0.31%) for the three test circuits among the three options for the LDE-aware layout migration. It is worth noting that, without using Algorithm 5, the number of the required simulations for LDE sensitivity analysis would be 2^{LParam}, where *LParam* is the number of the LDE parameters. In our experiments, the number of *LParam* is 20, 30, and 20 for the comparator, Cascode OPAMP, and two-stage OPAMP, respectively.

Table 6. LDE sensitivity analysis results for the Comparator, Cascode and two-stage OPAMP circuits

Circuits		Spec.	Regular Migration	LDE-Aware Migration		
				Min. α-Range	Avg. α-Range	Sat. Spec.
Comparator	*# Simulations*	-	-	115	62	48
	Area increment (%)	-	-	0.93	0.31	0.31
	Resolution (uV)	180	197	170	175	179
	Falling edge delay (ps)	25	24.43	24.41	24.2	24.18
	Rising edge delay (ps)	50	46.04	46.43	46.39	46.12
	Positive falling-edge overshoot (mV)	25	21.35	21.97	21.87	21.79
	Negative falling-edge overshoot (mV)	50	41.88	44.21	44.04	42.63
Cascode OPAMP	*# Simulations*	-	-	2581	275	66
	Area increment (%)	-	-	1.81	1.15	0.12
	DC Gain (dB)	50	48.04	50.55	50.38	50.14
	Gain Margin (dB)	50	53.35	53.32	53.26	53.38
	Phase Margin (degree)	75	76.77	75.16	75.41	75.28
	Unit Gain Band- width (MHz)	70	70.23	71.2	71.63	70.48
Two-Stage OPAMP	*# Simulations*	-	-	340	90	66
	Area increment (%)	-	-	0.38	0.38	0.11
	DC Gain (dB)	50	49.15	50.81	50.81	50.55
	Gain Margin (dB)	15	16.52	16.07	16.07	16.14
	Phase Margin (degree)	50	60.77	59.25	59.25	58.89
	Unit Gain Band- width (MHz)	500	501	521	521	515

As an important OPC metric, edge placement error (EPE) is defined as the difference between the original layout patterns and the final wafer image. A lower EPE value exhibits higher fidelity of layout image. For fair comparison with our proposed OPC-inclusive method,

we have implemented a state-of-the-art analog routing scheme [31] (entitled as "*Regular-Routing*" as the column header in Table 7), which considers a variety of analog constraints but without the capability of OPC handling. The EPE and the number of hot spots are reported in Table 7 between two methods for the three test circuits in our experiments.

Table 7. Experimental results of the regular routing and OPC-inclusive routing

Circuits		**Regular Routing** [31]	**OPC-Inclusive Routing**
Comparator	***EPE(um)***	20.70	4.55
	# Hot-Spots	130	0
Cascode OPAMP	***EPE(um)***	58.94	4.9
	# Hot-Spots	171	0
Two-stage OPAMP	***EPE(um)***	42.15	3.85
	# Hot-Spots	122	0

As shown in Table 7, EPE was decreased 4.5, 12, and 11 times for the comparator, Cascode OPAMP, and two-stage OPAMP respectively in our proposed OPC-inclusive router with no hot-spot left in the final layout compared to the regular routing method [31]. As discussed in Sub-section 5.3.2, if there is an unrouteable region after ILP-based router, one more operation called "Enlarging space among devices" (shown in Fig. 31) is needed to make space for the router to provide routing resources. Since this did not appear in all of our test circuits, in our experiments we tried to confine the number of routing layers to 2 for the Cascode OPAMP and two-stage OPAMP layouts to evaluate the capability of this proposed scheme for the situation of limited routing layer resources. Since the router was forced to route all the nets with only 2 layers but without generating any hot spots, there were two and one unrouteable incidents in the Cascode

OPAMP and two-stage OPAMP layouts, respectively. By using the space enlargement technique among the devices as described in Sub-section 5.3.2, all the nets were finally routed successfully. As a result of enlarging space among the devices, the total layout area was increased by 1.55% and 0.41% for the Cascode OPAMP and two-stage OPAMP, respectively.

5.5. Summary

In this chapter we have proposed a new LDE-aware layout migration methodology supported by an ILP-based OPC-inclusive routing algorithm for analog circuits. To address the challenging LDEs in the analog layout migration process towards the advanced nanometer technologies, we have included a smart sensitivity analysis that can identify the optimum solution to LDE parameters to gain higher circuit performance. A semi-dynamic-programming algorithm has been proposed to improve the efficiency of the sensitivity analysis, whose time complexity is originally in an exponential order. Moreover, an ILP-based OPC-inclusive router has been developed to route the layout by using the space reservation technique to remove any lithographic hot spots. Our experimental results show that the circuit performance and layout image fidelity can be largely improved for analog circuits as a result of LDE and OPC handling in our proposed methods.

Chapter 6 Conclusion

In this thesis, we have first explained the necessity of EDA for analog/RF circuits and then proposed novel methodologies to address multiple important challenges in the analog/RF EDA domain. As one of the physical design stages, routing, which aims to connect the device terminals, plays an important role in the layout synthesis. Analog/RF circuits have their special constraints much more than the routability constraint. It is normally the responsibility of the routing stage to ensure these nontrivial analog constraints are satisfied towards a successful tape-out. The state-of-the-art analog routing research tends to favor linear programming to achieve various analog constraints, which, although effective, fails to offer high routing efficiency on its own. We formulate the analog/RF routing problem in ILP formulation to simultaneously consider routability and analog/RF constraints. Our proposed method supports hierarchical routing, which can divide the entire routing area into multiple smaller heterogeneous regions where the ILP can efficiently derive routing solutions. We have developed a scheme to adaptively decrease routing resolution in the hierarchical routing to reduce the number of ILP variables and constraints to boost the efficiency of the proposed router. In addition, a white-space handling scheme has been developed to further improve the routing efficiency without compromising the routing quality, especially helpful for fairly larger analog and RF circuits. To avoid blind decision in decreasing routing resolution in hierarchical scheme, we have also developed an algorithm to predict the most suitable resolution decrement ratio values for the hierarchical levels. In addition, some performance-related parameters have been considered in the proposed ILP formulation to fit the analog/RF circuits and thus improve the performance. The experimental results show that our proposed adaptive ILP-based router, which can contribute to better performance of analog/RF circuits thanks to the performance-related

parameter consideration, is much faster than the conventional ones since it spends much less time for the areas that need no accurate routing anyway.

On the other hand, we have also illustrated that interconnect parasitics are making more impact on analog/RF circuit performance in the advanced technologies. Therefore, we have proposed an algorithm to address interconnect parasitics (mainly parasitic resistance and parasitic capacitance) in the routing flow to ensure the routing solution would not degrade the circuit performance. To achieve this end, the interconnect resistance and capacitance are extracted from the layout at the initial ILP routing stage and a sensitivity analysis stage would follow to find out an accurate range for the interconnect resistance and capacitance for the best circuit performance. Then we have developed a nonlinear programming (NLP) method to find the best interconnect width as per the interconnect resistance and capacitance. It actually means that the interconnect widths are treated as variables to yield the best circuit performance as the objective for the analog/RF layouts. In case there is a mismatch out of the routing results, it would be detected automatically in the global routing stage and our proposed mismatch minimization algorithm would be invoked to minimize such a mismatch for the sensitive matching nets. To the best of our knowledge, such a mismatch minimization problem, which cannot be handled by the other works previously published in the literature, has been defined and formulated in this thesis for the first time. To resolve the mismatch problem, we have proposed a smart A* algorithm to rearrange the Steiner points at the first stage and then evaluated the most suitable interconnect widths for the minimum mismatch by using a NLP formulation. Moreover, to address the electromigration (EM) issues that may cause interconnect short circuits and voids on metal vias, electromigration has been taken into account in our ILP formulation for the analog/RF routing besides the interconnect parasitic and mismatch minimization. The

experimental results have shown our proposed EM- and parasitic-aware analog router could reach the best circuit performance compared to the previous works while the highest MTTF was achieved due to the EM consideration in all the routing operations.

By using layout migration, an existing analog layout from an old technology can be retargeted to a new technology or migrated to new specifications while the layout design template remains the same in order to preserve the layout designers' skills in the target layout. According to the literature, analog circuit performance may be affected by LDEs and lithographic effects, which were not existing in the old technologies but increasingly more prominent in the advanced technologies. Therefore, the LDEs and OPC should be seriously addressed in the analog layout migration process. In this thesis, we have proposed an LDE-aware analog layout migration methodology supported by our ILP-based OPC-inclusive routing scheme. To include LDEs as one of the performance-related parameters in our proposed methodology, we have advocated to perform LDE sensitivity analysis to identify which LDE parameters can contribute to the optimum circuit performance in the migration process. Since the required simulations for the sensitivity analysis feature an extremely high (i.e., exponential) time complexity, we have developed a semi-dynamic-programming scheme that can improve the efficiency of sensitivity analysis based on the solution space and designer's demand. To maintain the layout template, a constraint graph representation has been used for the layout migration process. Moreover, OPC has been introduced as one of the most popular RET techniques in the sub-wavelength era, which is able to minimize the difference between ideal layout and the wafer image. We have proposed an OPC-inclusive routing method to take into account the lithographic constraints in the ILP routing. It can help guarantee that no hot spots will remain in the layout after the routing stage while the image fidelity can be also improved. An interconnect width

control formulation has been utilized to simultaneously address the OPC requirements along with the other routability/analog constraints. Since the interconnect width control formulation may need more routing resources, a space reservation scheme has been proposed to make extra space and thus provide more routing resources if needed. Similar to the LDE handling, the constraint graphs can help deal with any placement refinement that may be needed due to extra space adjustment. We have combined the LDE-aware layout migration and OPC-inclusive routing scheme together to provide a unified EDA tool for analog layout generation and optimization. The experimental results show that our proposed methods can not only improve the circuit performance as a result of the LDE handling, but also suppress lithographic hot spots in the layout with much better image fidelity compared to the previous works.

Chapter 7 Future Work

Since our proposed ILP formulation is capable of admitting new constraints, it is possible to extend such a formulation to address the other issues in the analog, RF, digital, and mixed-signal integrated circuit challenges. Fin Field Effect Transistors (FinFETs) have been introduced as one of the best alternatives in the modern technologies since conductive channel becomes shorter and it is more challenging to effectively control short-channel effects in the conventional CMOS technology nodes of 22nm and beyond. There are many open areas in the FinFET manufacturing physical design process that can be investigated as future work of this thesis. Below we will just discuss some of them briefly.

Gate misalignment in the lithography process can change the threshold voltage and drain current. Placement would also be critical due to the matching constraints in the analog FinFET layouts [83]. The idea proposed for layout migration in this thesis can be extended to address the distortion due to the gate misalignment and also placement refinement in the FinFET design. Since FinFET routing should respect the Multiple Patterning (MP) lithography design rules, a more sophisticated router is actually needed to guarantee all the constraints can be satisfied properly in the FinFET layouts. Unlike the conventional routing, multiple-patterning-aware routing has to route the electrical nets by using more than two different colors with no closeness by wires in the same color. Although several new works have appeared on digital multiple-patterning-aware routing, there is an obvious lack in the analog/RF domain. Recently, a common-centroid placement and routing method on analog FinFET layouts was proposed in [84]. As discussed earlier in this thesis, we believe our proposed router can be extended with more constraints to address multiple-patterning challenges for the FinFET layouts.

According to our experiments, placement solutions can definitely affect the routing results. We believe both CMOS and FinFET analog circuits are suffering from lack of powerful combined placement and routing tools. Thus, it would be interesting if placement and routing can be combined together to handle the analog/RF constraints in our future work. Although there were already some previous works in this area, the new challenges presented in this thesis will still need to have a comprehensive formulation to combine placement and routing algorithms. Parasitics, electromigration, LDE, OPC and multiple patterning would be new challenges that should be addressed by the combined placer-router. The parasitics should be taken into account by the combined placer-router since it is not sufficient to be handled sequentially by firstly placer and then router. Although electromigration is mainly the concern of a router rather than a placer, high current electrical nets require more routing resources. Therefore, it would be better to place them in the proper location to avoid any routing deficiency. The LDEs, on the other hand, are usually a placement concern, while a predictable scheme had better be established to identify a better placement solution that can lead to less wirelength for sensitive electrical nets. Recently, a new analog placement algorithm was proposed to satisfy LDE and other analog constraints in their formulation [57]. However, no routing aspects were considered in their formulation. It would be more practical if this work could be extended to an analog combined placer-router, which is able to handle the routing-related constraints. In this thesis we have already shown that the placement refinement is demanded in the space reservation operation for the OPC-inclusive routing in the layout migration process. Thus, a combined placer-router would be very helpful when OPC is targeted as the main objective.

In addition, the multiple-patterning-awareness, as an indispensable feature of the new routers for the advanced nanometer technologies, may suffer from limited coloring capability if

placement is fixed. The coloring operation, which is normally represented by constraint graphs, is to make sure the neighboring wires do not have the same color. At the moment most of the current work related to multiple patterning has been done for digital circuits, whereas there is very limited work in the analog domain. The constraint graphs for the coloring operation may be merged with that for the layout migration process proposed in this thesis. Thus, a multiple-patterning-aware layout migration method can be developed. To the best of our knowledge, no multiple-patterning-aware router has been proposed for the analog circuits. In particular, it is still unclear to what extent analog circuit performance can be affected due to multiple-patterning-related issues. Thus, a multiple-patterning-aware combined placer-router might be a good topic for the analog EDA physical design research especially towards the FinFET technology. Furthermore, since the structure of the FinFET devices is different from that of the conventional CMOS MOSFETs, layout migration from CMOS technologies to FinFET would be another interesting research topic as the future work.

References

[1] S. Sorkhabi and L. Zhang, "Automated topology synthesis of analog and RF integrated circuits: a survey," *Integration, the VLSI Journal*, vol. 56, pp. 128-138, 2017.

[2] P. H. Lin, H. C. Yu, T. H. Tsai and S. C. Lin, "A matching-based placement and routing system for analog design," in *Proc. International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, pp. 1-4, 2007.

[3] E. Yilmaz and G. Dundar, "Analog layout generator for CMOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 32–45, 2009.

[4] P. Khademsameni and M. Syrzycki, "A tool for automated analog CMOS layout module generation and placement," in *Proc. IEEE CCECE, Canadian Conference on Electrical and Computer Engineering*, pp. 416-421, 2002.

[5] M. P. H. Lin, Y. W. Chang and C. M. Hung, "Recent research development and new challenges in analog layout synthesis," in *Proc. 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, Macau, pp. 617-622, 2016.

[6] M. Strasser, M. Eick, H. Grab, U. Schlichtmann and F. M. Johannes, "Deterministic analog circuit placement using hierarchically bounded enumeration and enhanced shape functions," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, pp. 306-313, 2008.

[7] P.-H. Lin and S.-C. Lin, "Analog placement based on hierarchical module clustering," in *Proc. ACM/IEEE Design Automation Conference*, Anaheim, CA, pp. 50-55, 2008.

[8] P. Y. Chou, H. C. Ou and Y. W. Chang, "Heterogeneous B*-trees for analog placement with symmetry and regularity considerations," in *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, pp. 512-516, 2011.

[9] H. C. Ou, H. C. Chang Chien and Y. W. Chang, "Simultaneous analog placement and routing with current flow and current density considerations," in *Proc. 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX, pp. 1-6, 2013.

[10] L. Zhang, C. J. R. Shi and Y. Jiang, "Symmetry-aware placement with transitive closure graphs for analog layout design," in *Proc. Asia and South Pacific Design Automation*

Conference, Seoul, pp. 180-185, 2008.

[11] R. He and L. Zhang, "Symmetry-aware TCG-based placement design under complex multi-group constraints for analog circuit layouts," in *Proc. 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Taipei, pp. 299-304, 2010.

[12] L. Xiao and E. F. Y. Young, "Analog placement with common centroid and 1-D symmetry constraints," in *Asia and South Pacific Design Automation Conference*, Yokohama, pp. 353-360, 2009.

[13] Q. Ma, L. Xiao, Y. C. Tam and E. F. Y. Young, "Simultaneous handling of symmetry, common centroid, and general placement constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 1, pp. 85-95, 2011.

[14] S. Nakatake, "Structured Placement with Topological Regularity Evaluation," in *Proc. Asia and South Pacific Design Automation Conference*, Yokohama, pp. 215-220, 2007.

[15] S. Nakatake, M. Kawakita, T. Ito, M. Kojima, M. Kojima, K. Izumi, T. Habasaki, "Regularity-oriented analog placement with diffusion sharing and well island generation," in *Proc. 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Taipei, pp. 305-311, 2010.

[16] L. Zhang, Z. Liu, "Directly performance-constrained template-based layout retargeting and optimization for analog integrated circuits," *Integration, the VLSI Journal*, vol. 1, pp. 1-11, 2011.

[17] G. Shomalnasab and L. Zhang, "New analytic model of coupling and substrate capacitance in nanometer technologies," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 7, pp. 1268-1280, July 2015.

[18] L. Zhang, Y. Zhang, Y. Jiang and C. J. R. Shi, "Symmetry-aware placement with transitive closure graphs for analog layout design," *International Journal of Circuit Theory and Application*, vol. 38, no. 3, pp. 221-241, 2010.

[19] A. B. Kahng, S. Nath, and T. S. Rosing, "On potential design impacts of electromigration awareness," in *Proc. 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 527-532, 2013.

[20] E. Maricau and G. Gielen, Analog IC reliability in nanometer CMOS, New York, NY: Springer, 2013.

[21] I. A. Blech, "Electromigration in thin aluminum films on titanium nitride," *Journal of*

Applied Physics, vol. 47, no. 4, pp. 1203–1208, 1976.

[22] J. Xue, Y. Deng, ; Z. Ye, H. Wang, L. Yang and Z. Yu, "A framework for layout-dependent STI stress analysis and stress-aware circuit optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 3, pp. 498 - 511, 2012.

[23] T. Liao, L. Zhang, "Analog integrated circuit sizing and layout dependent effects: a review," *Microelectronics and Solid State Electronics*, vol. 3, pp. 17-29, 2014.

[24] J. Watts, K. Su and M. Basel, "Netlisting and modeling well-proximity effects," *IEEE Transactions on Electron Devices*, vol. 53, no. 9, pp. 2179-2186, 2006.

[25] M. V. Dunga, X.-M. (J.) Xi, J. He, W.-D. Liu, K.-Y. M. Cao, X.-D. Jin, J. J. Ou, M.-S. Chan, A. M. Niknejad and C.-M. Hu, "BSIM4.5.0 MOSFET model – user's manual," Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Chapters13-15, 2004.

[26] Li-Da Huang and M. D. F. Wong, "Optical proximity correction (OPC)-friendly maze routing," in *41st Design Automation Conference*, San Diego, pp. 186-191, 2004.

[27] C. A. Mack, Fundamental principles of optical lithography: the science of microfabrication, Chichester, U.K.: Wiley, 2007.

[28] M. Cho, K. Yuan, Y. Ban and D. Z. Pan,, "ELIAD: efficient lithography aware detailed routing algorithm with compact and macro post-OPC printability prediction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 7, pp. 1006-1016, 2009.

[29] J. Cohn, D. Garrod, R. Rutenbar, and L. Carley, "KOAN/ANAGRAM II: new tools for device-level analog placement and routing," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 3, pp. 330–342, 1991.

[30] L. Xiao, E. Young, X. He and K. Pun., "Practical placement and routing techniques for analog circuit designs," in *Proc. of International Conf. on Computer-Aided Design*, San Jose, CA, USA, 2010, pp.675-679.

[31] H.-C. Ou, H.-C. C. Chien, and Y.-W. Chang, "Nonuniform multilevel analog routing with matching constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1942-1954, 2014.

[32] P.-C. Pan, H.-M. Chen, Y.-K. Cheng, J. Liu, and W.-Y. Hu, "Configurable analog routing methodology via technology and design constraint unification," in *Proc. of the International*

Conference on Computer-Aided Design - ICCAD '12, pp. 620-626, 2012.

[33] Q. Gao, Y. Shen, Y. Cai and H. Yao, "Analog circuit shielding routing algorithm based on net classification," in *Proc. ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, Austin, TX, USA, 2010, pp. 123-128.

[34] K. Sajid, J. D. Carothers, J. J. Rodriguez and W. T. Holman, "Global routing methodology for analog and mixed-signal layout," in *Proc. 14th Annual IEEE International ASIC/SOC Conference*, Arlington, VA, 2001, pp. 442-446.

[35] J. Cortadella, J. Petit, S. Gomez, and F. Moll, "A boolean rule-based approach for manufacturability-aware cell routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 3, pp. 409–422, 2014.

[36] J. Cong, Jie Fang, Min Xie and Yan Zhang, "MARS-a multilevel full-chip gridless routing system," *in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 382-394, 2005.

[37] C. Chu and Y.-C. Wong, "FLUTE: fast lookup table based rectilinear steiner minimal tree algorithm for VLSI design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 70–83, 2008.

[38] G. Ajwani, C. Chu, and W.-K. Mak, "FOARS: FLUTE based obstacle-avoiding rectilinear steiner tree construction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 2, pp. 194–204, 2011.

[39] M. Ozdal and R. F. Hentschke, "An algorithmic study of exact route matching for integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 12, pp. 1842–1855, 2011.

[40] Q. Gao, H. Yao, Q. Zhou and Y. Cai, "A novel detailed routing algorithm with exact matching constraint for analog and mixed signal circuits," in *Proc. 12th International Symposium on Quality Electronic Design*, Santa Clara, CA, 2011, pp. 1-6.

[41] M. Ozdal and R. F. Hentschke, "Algorithms for maze routing with exact matching constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 1, pp. 101–112, 2014.

[42] H. Yao, Y. Cai, and Q. Gao, "LEMAR: A novel length matching routing algorithm for analog and mixed signal circuits," in *Proc. 17th Asia and South Pacific Design Automation Conference*, pp. 157-162, 2012.

[43] J. Lienig, "Electromigration and its impact on physical design in future technologies," in *Proc. of ACM international symposium on International symposium on physical design - ISPD '13*, pp. 33-40, 2013.

[44] S. Bigalke and J. Lienig, "Load-aware redundant via insertion for electromigration avoidance," in *Proc. of International Symposium on Physical Design - ISPD '16*, pp. 99-106, 2016.

[45] R. Martins, N. Lourenço, A. Canelas, and N. Horta, "Electromigration-aware analog router with multilayer multiport terminal structures," *Integration, the VLSI Journal*, vol. 47, no. 4, pp. 532–547, 2014.

[46] Baozhen Li, J. Gill, C. J. Christiansen, T. D. Sullivan and P. S. McLaughlin, "Impact of via-line contact on CU interconnect electromigration performance," in *Proc. IEEE International Reliability Physics Symposium*, San Jose, pp. 24-30, 2005.

[47] D. A. Li, M. Marek-Sadowska and S. R. Nassif, "A method for improving power grid resilience to electromigration-caused via failures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 1, pp. 118-130, 2015.

[48] Yi-Lung Cheng, S. Y. Lee, C. C. Chiu and K. Wu, "Back stress model on electromigration lifetime prediction in short length copper interconnects," in *Proc. IEEE International Reliability Physics Symposium*, Phoenix, pp. 685-686, 2008.

[49] I. H. R. Jiang, H. Y. Chang and C. L. Chang, "WiT: optimal wiring topology for electromigration avoidance," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 4, pp. 581-592, 2012.

[50] Y. C. Tsai, T. H. Li, T. C. Chen and C. W. Yeh, "Electromigration- and obstacle-avoiding routing tree construction," in *Proc. International Symposium on VLSI Design, Automation, and Test*, Hsinchu, pp. 1-4, 2013.

[51] J. M. Jonquères, J. M. Portal, G. Micolau and O. Ginez, "Current density aware algorithm for net generation in analog high current application," in *Proc. International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, Seville, pp. 153-156, 2012.

[52] G. Jerke, J. Lienig, J. Scheible, "Reliability-driven layout decompaction for electromigration failure avoidance in complex mixed-signal IC designs," in *Proc. IEEE Int. on Design Automation Conference*, San Diego, CA, USA, 2004, pp. 181-184.

[53] C.-C. Wang, W. Zhao, F. Liu, M, Chen and Yu Cao, "Modeling of layout-dependent stress effect in CMOS design," in *IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, San Jose, 2009, pp. 513-520.

[54] K.-W. Su et al., "A scaleable model for STI mechanical stress effect on layout dependence of MOS electrical characteristics," in *Proc. of the IEEE 2003 Custom Integrated Circuits Conference*, San Jose, pp.245 - 248, 2003.

[55] P.G. Drennan, M. L. Kniffin and D. R. Locascio, "Implications of proximity effects for analog design," in *Proc. Custom Integrated Circuits Conference*, San Jose, pp. 169- 76, 2006.

[56] S. Youssef, F. Javid, D. Dupuis, R. Iskander and M. Louerat, "A Python-based layout-aware analog design methodology for nanometric technologies," in *Proc. Design and Test Workshop (IDT), 2011 IEEE 6th International*, Beirut, pp. 62-67, 2011.

[57] H.-C. Ou, K.-H. Tseng, J.-Y. Liu, I.-P. Wu and Y.-W. Chang, "Layout-dependent effects-aware analytical analog placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 8, pp. 1243 - 1254, 2016.

[58] K. El-Kenawy and M. Dessouky, "Stress-aware analog layout devices pattern generation," in *Proc. Design & Test Symposium (IDT), 11th International*, Hammamet, pp. 233-238, 2016.

[59] I. Mohammed, K. El-Kenawy and M. Dessouky, "Layout dependent effects mitigation in current mirrors," in *Proc. Electronics, Communications and Computers (JEC-ECC), 2016 Fourth International Japan-Egypt Conference on*, Cairo, pp. 107-110, 2016.

[60] Y. Li, S.-M. Yu, and Y.-L. Li, "Intelligent optical proximity correction using genetic algorithm with model- and rule-based approaches," *Computational Materials Science*, vol. 45, no. 1, pp. 65-76, March 2009.

[61] J. Park, C. Park, S. Rhie, Y. Kim, M. Yoo, J. Kong, H. Kim, S. Yoo, S. Center, S. Cae, "An efficient rule-based OPC approach using a DRC tool for 0.18μm ASIC," in *Proc. of the 1st International Symposium on Quality of Electronic Design*, San Jose, pp. 81-85, 2000.

[62] X. Dong and L. Zhang, "Process-variation-aware rule-based optical proximity correction for analog layout migration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 8, pp. pp. 1395-1405, Aug. 2017.

[63] S. Banerjee, P. Elakkumanan, L. W. Liebmann and M. Orshansky, "Electrically driven optical proximity correction based on linear programming," in *Proc. IEEE/ACM*

International Conference on Computer-Aided Design, San Jose, pp. 473-479, 2008.

[64] R. Rodrigues, A. Sreedhar and S. Kundu, "Optical lithography simulation using wavelet transform," in *Proc. IEEE International Conference on Computer Design*, Lake Tahoe, pp. 427-432, 2009.

[65] J. Mitra, P. Yu, D. Z. Pan, "RADAR: RET-aware detailed routing using fast lithography simulations," in *Proc. of the 42nd annual Design Automation Conference*, Anaheim, pp. 369-372, 2005.

[66] S.-Y. Fang, C.-W. Lin, G.-W. Liao, Y.-W. Chang, "Simultaneous OPC- and CMP-aware routing based on accurate closed-form modeling," in *Proc. of the 2013 ACM International symposium on Physical Design*, Stateline, pp. 77-84, 2013.

[67] P. Y. Chou, N. C. Chen, M. P. H. Lin and H. Graeb, "Matched-routing common-centroid 3-D MOM capacitors for low-power data converters," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 8, pp. 2234-2247, 2017.

[68] Kuan-Hsien Ho, Hung-Chih Ou, Y. W. Chang and Hui-Fang Tsao, "Coupling-aware length-ratio-matching routing for capacitor arrays in analog integrated circuits," in *Proc. 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX, pp. 1-6, 2013.

[69] M. P. H. Lin, V. W. H. Hsiao, C. Y. Lin and N. C. Chen, "Parasitic-aware common-centroid binary-weighted capacitor layout generation integrating placement, routing, and unit capacitor sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 8, pp. 1274-1286, 2017.

[70] M. Torabi and L. Zhang, "Efficient ILP-based variant-grid analog router," in *Proc. 2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, QC, pp. 1266-1269, 2016.

[71] "MOSEK optimization software https://www.mosek.com/products/mosek," [Online].

[72] M. Torabi and L. Zhang, "A Fast Hierarchical Adaptive Analog Routing Algorithm based on Integer Linear Programming," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, no. 3, 2017.

[73] Z. Liu and L. Zhang, "Performance-constrained template-driven retargeting for analog and RF layouts," in *Proc. ACM Great Lakes Symposium on VLSI*, pp. 429-434, 2010.

[74] M. Torabi and L. Zhang, "Electromigration and Parasitic-Aware ILP Router for Analog and RF Circuits," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*,

Florence, Italy, pp. 1-5, 2018.

[75] G. Shomalnasab, H. Heys, and L. Zhang, "Analytic modeling of interconnect capacitance in submicron and nanometer technologies," in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 2553-2556, 2013.

[76] "http://www.i2c2.aut.ac.nz/Wiki/OPTI/index.php," [Online].

[77] M. Torabi and L. Zhang, "Electromigration- and Parasitic-Aware ILP-Based Analog Router," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* 2018.

[78] L. Zhang, N. Jangkrajarng, S. Bhattacharya, and C.-J. R. Shi, "Parasitic-aware optimization and retargeting of analog layouts: a symbolic-template approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 27, no. 5, pp. 791–802, 2008.

[79] X. Dong and L. Zhang, "Lithography-aware analog layout retargeting," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 35, no. 2, pp. 232-245, 2016.

[80] H. S. Yang et al., "Dual stress liner for high performance sub-45nm gate length SOI CMOS manufacturing," in *Proc. IEDM Technical Digest. IEEE International Electron Devices Meeting,* San Francisco, pp. 1075-1077, 2004.

[81] X. Ma and G. R. Arce, Computational Lithography, Hoboken, NJ, USA: Wiley, 2010.

[82] "Mentor-Graphics Calibre nmOPC.," in *[Online]. Available: https://www.mentor.com/products/ic-manufacturing/.*

[83] PH Wu, MPH Lin, X Li, TY Ho, "Common-centroid FinFET placement considering the impact of gate misalignment," in *Proc. of the 2015 Symposium on International Symposium on Physical Design*, Monterey, California, USA, pp. 25-31, 2015.

[84] PH Wu, MPH Lin, X Li, TY Ho, "Parasitic-aware common-centroid FinFET placement and routing for current-ratio matching," *ACM Transactions on Design Automation of Electronic Systems (TODAES),* vol. 21, no. 3, pp. 1-22, 2016.

Appendix: Published Papers and Prepared Manuscript

[1] M. Torabi and L. Zhang, "A Fast Hierarchical Adaptive Analog Routing Algorithm based on Integer Linear Programming," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, No. 3, Article 55, 2017.

[2] M. Torabi and L. Zhang, "Electromigration- and Parasitic-Aware ILP-Based Analog Router", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2018.

[3] M. Torabi and L. Zhang, "Efficient ILP-Based Variant-Grid Analog Router," *in IEEE International Symposium on Circuits and Systems*, May 2016, pp. 1266-1269.

[4] M. Torabi and L. Zhang, "Electromigration and Parasitic-Aware ILP Router for Analog and RF Circuits," *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, Italy, 2018, pp. 1-5.

[5] M. Torabi and L. Zhang, "An algorithm for symmetrically exactly matched obstacle avoiding routing," *in Newfoundland Electrical and Computer Engineering Conference (NECEC)*, St. John's, Canada, 2014.

[6] M. Torabi and L. Zhang, "LDE-Aware Analog Layout Migration with OPC-Inclusive Routing", Submitted to *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.