CAD for Analog/Mixed-Signal Integrated Circuits

Mohamed B. Alawieh, Ahmet F. Budak, Hao Chen, Mingjie Liu, David Z. Pan†, Wei Shi, Xiyuan Tang, Shuhan Zhang, Keren Zhu
dpan@ece.utexas.edu
Department of Electrical and Computer Engineering
The University of Texas at Austin

Abstract

While digital integrated circuits (ICs) has adopted highly automated computer aided design (CAD) tools for decades including synthesis, placement, and routing, analog and mixed-signal IC designs still largely rely on manual efforts. Due to increasing demands for high-performance, low-power and short time-to-market, analog design automation has received increasing interests and demand from both industry and academia. This chapter provides an overview on recent advancements in analog and mixed-signal IC design automation. We will cover key steps of the analog design flow, including system architecture definition, circuit topology selection, device sizing, layout design, and post-layout verification.

I. Introduction

Integrated circuits (IC), the backbone of modern electronics, have played a critical role in the significant advancement of modern information technology in the past decades. The ever-increasing level of integration complexity, now featuring multi-billion-transistor ICs, has powered numerous complicated tasks covering cloud computing, artificial intelligence, and smart manufacturing. Such complete systems now have been integrated on a few chips or even one single chip. One example of such Systems on Chip (SoC) can be found in the wireless communication chips in smartphones, including analog, digital, and radio-frequency (RF) sections on a single chip.

As technology nodes scale into the nanometer era, digital circuits have benefited the most, with significantly boosted computing power. Despite the trend to replace analog functions with digital circuits, e.g., shifting the traditional analog channel selection into digital signal processing, a few typical functions will always remain analog, including:

1) Interfaces between the digital chip and the real world: the physical world is analog and relies on analog/mixed-signal (AMS) components to interact. On the input side of a system, the specialized sensor captures physical signal (e.g., sound, temperature, and magnetic) and converts it into electrical representations such as voltage and current. A typical signal chain includes a low-noise amplifier, a filter, and an analog-to-digital converter (ADC). High-quality designs require a sufficient signal-to-noise-and-distortion ratio (SNDR). On the output side, the digital signal needs to be converted back to analog form to interact with the environment.
2) Power management components: with the increased integration level of modern electronic devices, a single system usually requires multiple power rails. For instance, LED drivers, logic units, and power amplifiers operate under drastically different voltage supplies. Power management units, required in all modern electronics, offer high-efficiency and compact voltage conversion for various functional blocks.
3) Clock generation circuits: like supply voltages, a modern electronic system requires multiple clock frequencies for sub-blocks. Derived from a standard reference block, commonly a crystal oscillator, the frequency synthesizer generates a range of frequencies required in SoC. Standard blocks include oscillators, charge pumps, and phase-lock loops.
4) Communication blocks: the colossal data demand in the internet of things (IoT) era poses new challenges to communication blocks. The emerging communication standards, e.g., 5G, machine-type communication, keep pushing boundaries of circuits' speed and energy efficiency. Standard blocks include transmitters and receivers. They are often referred to as radio-frequency (RF) circuits.
5) Highly customized mixed-signal chips targeting cutting-edge performance: for example, the state-of-the-art memory cells are mainly custom designed like analog circuits to push speed or power limit.

Clearly, analog, mixed-signal, and RF circuits are indispensable in modern electronics systems, especially those interacting with the real world. The emerging applications, including IoT, 5G networks, advanced computing, and healthcare electronics, pose sharply increasing demand for analog, mixed-signal, RF circuits. For example, as forecasted by Semiconductor Research Corporation (SRC), more than 45 trillion sensors will appear worldwide by 2032 [1]. Therefore, a short turnaround time of analog, mixed-signal, RF IC design is highly desired.

Despite the continuous efforts in analog layout automation, those accomplishments have not been well adopted in current industrial flows. The main reason is rooted in the characteristics of AMS circuits, which are complex and sensitive. Compared to its digital counterpart, AMS design deals with a wide range of specific circuit classes, various device types and often requires

† Corresponding author.

customization for each circuit type. Besides, they are sensitive to signal couplings, layout effects, and fabrication variations. Furthermore, the lack of effective ways to model the layout effects on analog performance imposes significant challenges for automation tools. Therefore, implementing analog circuits is still mainly a manual, time-consuming, and error-prone task. Typical analog design procedure includes:

1) System architecture definition: in a top-down design methodology, the designers first need to translate a system-level requirement into detailed sub-block specifications. For example, in an RF receiver design, the system architect should break down the sensitivity requirement into LNA noise figure, amplifier gain, and ADC signal-to-noise ratio.
2) Circuit topology selection: based on the sub-block specifications, a specific circuit topology is chosen. For instance, if a wide-band design is required in an OTA design, the feed-forward compensated structure is preferred. If a high gain is needed, the telescopic amplifier is preferred.
3) Device sizing: with the selected topology, devices, including transistors, capacitors, and resistors, need to be appropriately sized for the targeted performance. Due to the extensive exploration space, this progress is extremely challenging. For instance, in a simple 2-stage miller compensated OTA design, over 40 parameters need to be determined, such as transistor widths, lengths, biasing currents, voltages, and capacitor values.
4) Layout design: unlike digital circuits, which only need to satisfy design rule check (DRC), AMS layout generation faces extra challenges due to its high sensitivity. Constraints such as symmetry and critical nets directly impact the layout quality. Various layout-dependent effects, such as length of diffusion (LOD) and well proximity effect (WPE), significantly increase layout generation complexity.
5) Post-layout verification: post-layout extraction captures parasitics and layout effects. To ensure the layout quality, comprehensive verifications, including process-voltage-temperature (PVT) simulations and Monte-Carlo simulations, are performed.

Note that due to the nature of analog design, trade-offs are considered in every single step. For instance, one can hardly size an OTA to realize low power, high gain, and high BW simultaneously. Hence, AMS design depends highly on the application requirement and requires customization. Besides, multiple iterations are expected to meet the final specification.

This chapter aims to review the Computer Aided Design (CAD) developments dedicated to each step of the analog design flow mentioned above and provide a helpful guide to CAD designers who are interested in improving the design efficiency of analog, mixed-signal, and RF circuits.

II. Front-End CAD

Analog/mixed-signal Integrated Circuit design is a complex process involving multiple steps. Billions of nanoscale transistor devices are fabricated on a silicon die and connected via intricate metal layers during those steps. The final product is an IC, which powers much of our life today. Today, many systems require both digital functional units and analog subsystems to be implemented on the same chip. Thanks to advances in Electronic Design Automation (EDA) tools, most of the digital units can be efficiently and reliably synthesized. On the other hand, AMS design lacks these tools and is mostly handcrafted by analog specialists. Therefore, AMS design suffers from long design cycles and high design complexity and often constitutes a bottleneck in mixed analog/digital design [2]. Front-end CAD for AMS design includes topology selection and device sizing. Both these phases are considered costly since topology selection relies on the designer's experience on various architectures and device sizing is manual and conducted in an iterative manner. Therefore, front-end design consumes a significant portion of the total design budget. In a typical AMS design cycle, a designer starts with a schematic design of a circuit topology. The designer either selects a pre-existing topology based on circuit performance specifications or designs an entirely new topology based on domain knowledge. Usually, a circuit is expected to meet specifications (e.g., phase, delay, gain, power, bandwidth, etc.) at tens of operation corners. If specifications are not met at any specific corner, circuit designers modify the topology or resize circuits. Circuit resizing involves modification of device parameters like widths and lengths of MOS transistors used in the circuit. Due to the complex nature of circuits, it may not be obvious which devices impact specifications. This usually leads to trial and error, along with over design, in circuit sizing to meet performance metrics. After performance specifications are met at all corners, variation/yield/aging/reliability simulations are performed to ensure design functionality in the presence of process and usage scenarios. If a circuit fails to meet specifications at this stage, the designer has to resize the circuit again. Hence, AMS front-end is highly iterative and manual in nature. In order to tackle this labor-intensive nature and reduce time-to-market requirements, front-end automation for AMS design has attracted high interest in recent years.

A. Circuit Architecture and Topology Design Space Exploration

As it was mentioned in introduction, current practice in circuit topology selection relies on designer's expertise. After getting system requirements, designers use their understanding of circuit behaviors, in-depth analytic equations and technology performance to choose the most appropriate topology. Automation in topology exploration is desired due to various reasons. It is an important determinant of the final circuit performance and the first part of the whole flow; hence, automating this part is necessary to have a complete automated design flow since topology selection and device sizing are generally considered jointly. Despite the hard nature of topology selection problem, there have been some efforts to automate this phase. Some methods

realized the nature of human design flow and have been developed based on heuristics and design knowledge [3]–[5]. On the other hand, other methods targeted topology selection problem with machine learning solutions [6], [7]. In [3], a DRC clean layout of an OPAMP is designed from a system level specification. The topology selection is pruned via a decision tree whose branching is determined by the application purpose and performance requirements. Similarly, in [4], a method using fuzzy-logic based decision rules to determine circuit topology is adapted. The fuzzy-logic translates the expert defined design rules into a decision mechanism. Then the topology decision process is conducted via a defuzzification interface. However, as the technology advances and circuits become more complex, design knowledge is difficult to be extracted, collected and reused. Data-driven techniques are investigated to automate the topology selection with simulation results. In [6], a Convolutional Neural Network (CNN) is used as a classifier with training dataset containing circuit specifications as the input and topology index as the output. After being trained, the CNN can select the best topology given the specifications. The main challenge is the time-consuming process of obtaining performance data of different topologies. Also, topology data-base is usually limited in prior works which makes it a challenge to easily adapt it for larger domains.

It is worth noting that aforementioned studies choose one topology within a fixed database. In addition to these, some efforts to generate circuit topology from the performance requirements is shown. These studies are generally restricted to certain type/purpose of circuits and generalizing topology generation to various applications is currently untouched research area. [5] presents one of the first attempts on the area where genetic programming is used as the optimization engine for topology generation. A recent study introduces a hypernetwork scheme specialized to determine alignment, type and value of the components of two-port networks [7]. Since this method is proposed for specific type of circuits (two-port networks), adapting it for other AMS design applications may not be direct and could require further customization.

B. Device Sizing

AMS device sizing task stands for finding optimal values of design parameters to maximize the circuit performance metric. Although the majority of digital design flow is automated by EDA tools, AMS design practice is still heavily manual. In general, AMS design variables, such as transistor lengths & widths, capacitor values, biasing parameters etc., are adjusted based on designer's knowledge and experience [8]. However, due to the downscaling in the technology node and tightening time-to-market requirements, it gets more challenging for designers to develop state of the art, robust AMS blocks while accounting for more complex device models and severe process variations. Therefore, the need to bring automation solutions for AMS circuit sizing is obvious.

AMS Circuit Sizing: Problem Formulation

In general, AMS circuit sizing problem can be formulated in two ways. The first way is to formulate it as a constrained optimization problem succinctly as below.

$$\begin{aligned} &\text{minimize} \;\; f_0(\mathbf{x}) \\ &\text{subject to} \; f_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \ldots, m \end{aligned}$$

where, $\mathbf{x} \in \mathbb{D}^d$ is the parameter vector and d is the number of design variables of sizing task. Thus, $\mathbb{D}^d$ is the design space. $f_0(\mathbf{x})$ is the objective performance metric we aim to minimize and $f_i(\mathbf{x})$ is the i^{th} performance constraint in the design. The second way of formulating sizing problem is to transform it into an unconstrained problem by defining a Figure of Merit (FoM) and optimizing for it. In this way, circuit performance values are lumped into a single equation by using normalization constants, w.

$$\text{minimize} \;\; \text{FoM} = \sum_{i=0}^{m} w_i \times f_i(\mathbf{x})$$

Methods for AMS Circuit Sizing

Prior work on analog circuit sizing automation can be divided into two categories: *knowledge-based* and *optimization-based methods*. In the knowledge-based approach, design experts transcribe their domain knowledge into algorithms and equations [9], [10]. However, such methods create dependency on expert human-designers, circuit topology, and technology nodes. Thus, these methods are highly time-consuming and not scalable.

Optimization-based methods are further categorized into two classes: *equation-based* and *simulation-based* methods [11]. Equation-based methods try to express circuit performance via posynomial equations or regression models using simulation data. Then the equation-based optimization methods such as Geometric Programming [12], [13] or Semidefinite Programming (SDP) relaxations [14] are applied to convex or non-convex formulated problems to find an optimal solution. These methods are generally fast but due to scaling in technology and advances in circuit topologies developing accurate expressions for circuit performances is not easy and may deviate largely from the actual values. Therefore, the most recent trend is to employ simulation-based methods to tackle the sizing problem [15]–[29]. These methods make guided exploration in the search space and target a global minimum using the real evaluations from circuit simulators. In general, black-box or learning-based optimization techniques are applied to explore design space.

Traditionally, there have existed various model-free optimization methods such as particle swarm optimization (PSO) [30] and advanced differential evolution. [31] is proposed for analog circuits and [32] is proposed for RF where evolutionary strategies and

simulating annealing are used as the search mechanisms, respectively. Although these methods have good convergence behavior, they are known to be sample-inefficient (i.e., SPICE simulation intensive). Recently surrogate model-based and learning-based methods are becoming increasingly popular due to their efficiency in exploring solution space. In surrogate model-based methods, Gaussian Process Regression (GPR) [33] or Artificial Neural Networks (ANN) [34] are generally used for design space modeling, and the next design point is determined through model predictions. For example, MMLDE [35] method is proposed for synthesizing passive components of high-frequency Radio Frequency (RF) ICs. MMLDE is a hybrid optimization scheme where the global search is conducted via evolutionary steps and surrogate-model is build to make selections among candidate solutions. In MMLDE, authors used both GPR and ANN as the surrogate. GASPAD method followed this work and is introduced into Radio Frequency (RF) IC synthesis where GPR predictions guide evolutionary search [36]. WEIBO method proposed a GPR based Bayesian Optimization [37] algorithm where a blended version of weighted Expected Improvement (wEI) and the probability of feasibility is selected as acquisition function to handle constrained nature of analog sizing [21], [38]. The main drawback of Bayesian Optimization methods is its scalability as GP modeling has cubic complexity in the number of samples, $\mathcal{O}(N^3)$. An attempt to overcome this is given in [39], where authors utilized a neural network to make bayesian inference which eliminated the cubic dependency on the number of samples.

Other model-based optimization methods utilized ANN for circuit performance modeling. In [40], ANN is used as a proxy for the circuit simulator and performance estimates obtained by the trained model are used to efficiently guide an evolutionary optimization framework. [41] is also an ANN boosted evolutionary method, they used the trained ANN as a performance classifier between the parent population and child population and they modified the exploration procedure for only allowing promising child solutions to survive into next generation. In [42], authors proposed to enhance the efficiency of a Genetic Algorithm based optimizer where the the local minimum search is conducted via the help of trained ANN. Another method that utilizes neural nets is ESSAB [26] where ANN training is combined with a data augmentation method to reduce the need for required number of samples for accurate training. A different approach is adapted for building their ANN model in [43], where they use circuit specifications at the input nodes and design variables are taken from the output nodes. To train and work with such model a large dataset is generated for offline training of the ANN model.

Recently, reinforcement learning (RL) algorithms are applied in the area as learning-based methods. GCN-RL [27] leverages Graph Neural Networks (GNN) and proposes a transferable framework. They employed actor-critic algorithms to optimize circuit performance and circuit FoM is used as the reward value for training. They also demonstrated a transfer mechanism between topologies and technology nodes by modifying the state vectors. It reports superior results over various methods and human-designer and also shows successful transfer of learning between technology nodes. Drawbacks of this method is that it requires thousands of simulations for convergence (without transfer learning) and it suffers from engineering effort to determine observation vector, architecture selection, and reward engineering. AutoCkt [28] is a sparse sub-sampling RL technique optimizing the circuit parameters by taking discrete actions in the solution space. AutoCkt shows better efficiency over random RL agents and Differential Evolution. Since it adapts an offline training scheme, it requires to be trained with thousands of SPICE simulations before deployment, which is a sink cost for using their algorithm. Another RL method is DNN-Opt [29] where the core algorithm is inspired from deep RL actor-critic methods but customized for AMS sizing task. It offers an efficient online training methodology and shows strong performance convergence both in terms of number of required samples and total optimization time. Further, it introduces a recipe to apply their method to large-scale industrial circuits.

III. Layout Automation

In current AMS and RF integrated circuit (IC) design flows, the physical layout implementation stage is still heavily manual, thus time-consuming and error-prone, setting limits on the turnaround time. The main reason is originated in the high complexity of the AMS/RF IC design problem itself. Compared to its digital counterpart, AMS design is much more complicated even for simple modules as it considers various primitive devices and a wide range of diverse circuit classes that require customized tuning. Sensitive signal couplings, layout-dependent effects, and process variations of AMS/RF layouts also impose further challenges toward high-quality layout design.

Though the design automation techniques for AMS and RF circuits are not yet mature enough for standardized commercial applications, recent research has achieved promising advancements. AMS Layout automation techniques can be categorized into two paradigms: procedural layout generation and optimization-based layout synthesis [44]. Procedural layout generators utilizing pre-designed parameterized layout templates to migrate layouts for various manufacturing technologies and device sizings have been demonstrated. Optimization-based techniques using place-and-route (P&R) algorithms to optimize the area, power, and certain performance metrics have also been developed. These approaches possess various degrees of generality and are designed for different usage scenarios. Similar paradigms are also being utilized in automating RF layout synthesis. This section introduces the two paradigms and reviews some techniques utilizing them.

A. Procedural Layout Generation

Procedural-based layout generation refers to using pre-designed procedures to generate the layouts. It follows the procedures or templates to the layout structures. The manually designed templates enable sophisticated and high-quality layouts at the cost of extensive human efforts. Figure 1 shows a typical flow.

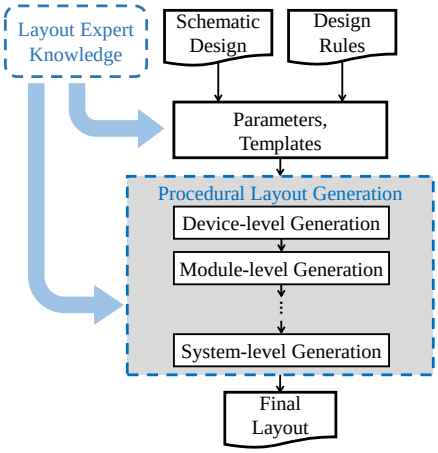

Fig. 1: Procedural analog circuits layout design flow [44].

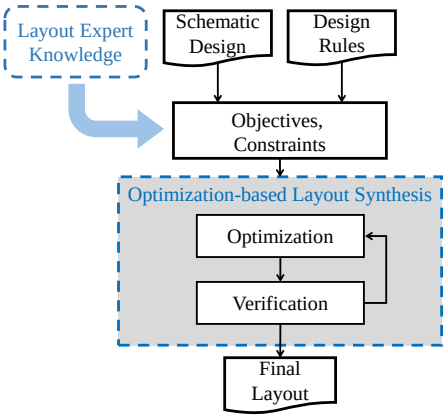

Fig. 2: Optimization-based layout design flow [44].

The idea of leveraging manually optimized templates to generate new layout traces back to VLSI design automation's early days. In early efforts in automating analog layout generation, such as ILAC [45] and SLAM [46], Procedural block generators are proposed to generate critical circuit substructures, such as differential pair and capacitors, with parameterizable templates. Later literature extends the procedural layout generation to support more sophisticated functionalities and reduce human efforts. Youssef et al. [47] developed a Python-based layout generation tool for generating primitive blocks. Han et al. [48] developed a GUI-based template engine to assist the development of layout templates. Some studies propose applications leveraging procedural analog layout generation. Stefanovic et al. [49] and Lopez et al. [50] apply procedure-based analog layout generation for parasitic estimation and device sizing. Ding et al. [51] combine template-based layout tool with digital P&R flow to automating SAR ADC layout synthesis. Wulff and Ytterdal [52] also demonstrates success in automatically generated SAR ADC layout with a Perl script-based layout compiler. Procedural layout generator also apply to RF IC demonstrated in [53].

To create a sustainable eco-system and enable broader reuse of design templates, people are proposing comprehensive standards. A notable attempt is the series of Berkeley Analog Generator (BAG). BAG [54] and BAG2 [55] propose a Python-based interface to enable programming and parameterizing the procedures for generating layouts. Leveraging the flexibility of programming language, BAG2 provides enhanced capability to contain well-optimized layout generation procedures. BAG2 have demonstrated its effectiveness in silicon with generating a SAR ADC [56] and analog parts in some other designs [57], [58]. The standardized procedural layout generation can benefits the parameterization and technology mitigation for the A MS circuit design by reusing the layout templates. It still requires manual tunning in designing and programming the layout procedures for each schematic.

B. Optimization-based Layout Synthesis

Optimization-based layout synthesis aims to provide highly automated layout solutions for AMS circuits. Similar to the digital IC physical design flow, it usually divides the AMS layout synthesis into several stages. Each stage is then formulated

as a constrained optimization problem. The sensitive analog layout considerations are indirectly handled by formulating them into objectives and constraints.

Figure 2 shows a typical optimization-based layout synthesis flow. As shown in the flow chart, the inputs of an optimization-based synthesis flow require minimum human effort. Only circuit netlists and technology specifications are necessary. The core of an optimization-based layout flow consists of four main subtasks: constraint extraction, module generation, placement, and routing. The computation flow is widely adopted in existing optimization-based frameworks like ALIGN [59] and MAGICAL [60], [61].

The constraint extraction step comprehends the layout design conventions by analyzing the pre-layout circuit information. For instance, the sensitiveness of pairing nets, interconnections between devices and building blocks, and the geometry structure of each module. As the extracted constraints can largely impact the final layout and the post-layout performance, generating accurate constraints for optimization-based layout design methodology is crucial. [62]. Typically, design conventions are realized by two major types of constraints: geometrical constraints and electrical constraints. Geometrical constraints implicitly handle some specific design aspects such as differential net current balancing by restricting the relative positions of some modules and nets. Common geometrical constraints include symmetry matching, common-centroid, proximity, etc [62]–[64]. The survey paper [65] gives an overview on the recent advancements in constraint extraction for geometrical constraints. Electrical constraints prevent circuit performance degradation caused by IR-drop, signal coupling, and parasitic coupling. Some widely adopted electrical constraints are minimum metal width, minimum via cuts, and net shielding [66].

Module generators provide the layout for the low-level modules, such as transistors and resistors. Primary primitive cell generators are widely available in commercial tools and process design kits (PDKs). On the other hand, some academic frameworks have customized device generators for open-source purpose [59], [61].

Placement determines the physical location of each generated module. Analog placement is usually formulated as a constrained optimization problem, while essential layout considerations are handled through imposing constraints. Symmetric constraint, which restricts some modules to be placed along the symmetric axis to reduce mismatch, is widely used [67]–[98]. Common-centroid is another widely adopted constraint [78], [80], [81]. Other constraints have also been proposed, including array structure in layouts [82], monotonic power current flow [90], [91], [96], thermal effects [76], system signal flow [99], and well proximity effects [100]. In addition to AMS circuits, symmetric constraint is also applied in RF IC layout synthesis [101].

The routing stage implements the interconnection with metal wires and VIAs. Analog routing is often formulated into optimizing wirelength while satisfying the symmetry between net pairs [102]–[110]. Ou et al. [105] relaxes the restricted symmetric constraints into different levels, namely symmetry, common-centroid, topology, and proximity. The exact matching constraint is also proposed as an alternative, where the length of wires on each metal layer is constrained to be matched [111], [112]. There are other works that forbid routing over the active regions of transistors [107], [113], optimize power routing [114], [115] and propose shielding critical nets [110]. Recently, Chen et al. [116] propose optimizing the total symmetry across the design. They present a new algorithm to match the layouts' pins and route the nets in symmetry instead of relying on given symmetric constraints. A higher degree of symmetry is found beneficial to circuit performance.

A rising trend in optimization-based AMS circuit layout automation is to create end-to-end frameworks [117]. ALIGN [59] adopts a grid-based methodology. The primitive generation, placement, and routing are all aligned with the grid, and design rules are handled from Correct-by-construction. On the other hand, MAGICAL [61] allows more flexibility to target human-like layout style. Equips with end-to-end flow consisting of module generator, analog placer, and detailed router, both ALIGN and MAGICAL have demonstrated success in automating building block-level analog circuits with low or no human efforts. MAGICAL further adopts the mixed-signal placement techniques [99] and state-of-the-art analog detailed router [116]. As a result, MAGICAL recently proves its effectiveness in silicon with a high performance $\Delta\Sigma$ ADC [61].

In summary, optimization-based layout synthesis requires the least human effort in the three approaches. However, it does not directly generate the layout from human expertise and requires a carefully formulated optimization problem. On the other hand, constraints to the automated tools still need manual input. To further improve the quality and reduce human efforts for the optimization-based AMS circuit layout automation, recent studies explore various emerging techniques.

Several studies focus on optimizing the post-layout performance beyond conventional constrained optimization settings. Xu et al. [118] propose to generate the wells similar to human practice. The proposed framework uses a generative adversarial network (GAN) to automatically learn how experienced engineer draws the wells and apply the learned pattern in automated flow. It is also extended to become part of the placement process in [100]. Zhu et al. [119] apply a similar idea to analog routing. Instead of relying on conventional constraints, the proposed router uses a variational autoencoder (VAE) to learn where human engineers route the special nets and use the machine learning model to guide the automated detailed router. In addition to mimicking the human implementations, people are proposing techniques to directly optimizing the performance. Liu et al. [120] explores the layout-to-performance modeling problem. The proposed machine learning model applies a convolutional neural network (CNN) to predict the post-layout performance from the placement results. Li et al. [121] integrate the performance prediction model in automated placement. The proposed framework treats the performance prediction as to the placement problem's objective to enable direct optimization on performance.

IV. Post-Layout Extraction and Verification

A clean DRC report shows that the layout abides by all the design rules for the technology node such as metal width, minimum spacing, etc. However, these checks do not guarantee that the layout corresponds to the desired schematic level design. Hence, a critical step at this stage is to extract a post-layout netlist to be compared against the schematic level netlist. Using commercial tools, a post-layout extraction step is performed where all components in the design, including connectivities and parasitic, are extracted to form a post-layout netlist. In general, resistance and capacitance parasitic are the main focus of the extraction step. However, in analog/RF designs inductance extraction for interconnect is also needed.

Given both netlists at the schematic and post-layout levels, Layout Vs. Schematic (LVS) check is performed to ensure that the layout corresponds to the schematic level design. A clean LVS check indicates a correct correspondence between layout and schematic. At this stage, the layout corresponds to the circuit intent at the schematic level while abiding by all design rules.

Despite these checks, the circuit may not be design ready at this point. In fact, the layout quality has a significant impact on the circuit performance especially in AMS designs. The layout extraction step generates a new netlist that includes layout-dependent parasitic. Hence, it is imperative that the new netlist is verified through simulation to make sure that, given the layout induced parasitic, the design still meets the requirements.

A final step in the design process is to assess the reliability of the AMS design against variation in the manufacturing process. Two major assessment are necessary [122]. The first is evaluating the performance under different operating conditions; mainly process, voltage and temperature (PVT) corners that the chip may face after fabrications. This analysis is application dependent since the fields of use are different for different chips, and so are the requirements. Therefore, PVT analysis is performed to ensure the correct operation of the design under the conditions it is meant to encounter.

While PVT corners can be determined in most cases, the design should also be simulated under random process variations. Examples of such variations include change in oxide thickness (Δ_{tox}) and random dopant fluctuations (RDF), among others. In fact, with the continuous scaling, process variations manifest itself among the most prominent factors limiting the yield of analog and mixed-signal (AMS) circuits [122], [123]. To quantify the impact of this variation, the design is simulated under different realizations of this variation [122], [123]. Using Monte Carlo (MC) analysis, a set of points are sampled from the variation distribution and circuit performance is performed. With such analysis, it is possible to evaluate the expected parametric yield of the manufactured chips, i.e., the percentage of manufactured chips that meet the design requirements. It is important to note though that MC analysis is computationally expensive, and achieving high confidence estimation for modern designs may require thousands of simulations. In literature, different techniques have been proposed to reduce the computational cost of MC analysis, e.g., using importance sampling and scaled sigma sampling [124], [125].

V. Conclusion

Although most of the digital circuits can be efficiently and reliably designed with advanced electronics design automation tools, analog and mixed-signal circuits still heavily rely on manual design. This book chapter reviews the recent advancements in CAD techniques and covers key steps in analog/mixed-signal circuit designs flow, including system architecture definition, circuit topology selection, device sizing, layout design and post-layout verification. Large efforts have been made to apply the most advanced machine learning techniques to reduce the burden of human in the loop and further automate the analog/mixed-signal circuit design process. Due to the rapid advancements of machine learning, future analog and mixed-signal IC designs will significantly leverage both human and machine intelligence for quality of results and design productivity.

ACKNOWLEDGEMENT

This work is supported in part by the NSF under Grant No. 1704758, and the DARPA IDEA program.

References

[1] "The Decadal Plan for Semiconductors," https://www.src.org/about/decadal-plan/, accessed: 2021-01-30.

[2] P. R. Gray, "Analog ic's in the submicron era: Trends and perspectives," in *1987 International Electron Devices Meeting*, 1987, pp. 5–9.

[3] H. Y. Koh, C. H. Sequin, and P. R. Gray, "Opasyn: a compiler for cmos operational amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 2, pp. 113–125, 1990.

[4] J. Chavez, A. Torralba, and L. G. Franquelo, "A fuzzy-logic based tool for topology selection in analog synthesis," in *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, vol. 1, 1994, pp. 367–370 vol.1.

[5] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, *Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming*. Dordrecht: Springer Netherlands, 1996, pp. 151–170.

[6] T. Matsuba, N. Takai, M. Fukuda, and Y. Kubo, "Inference of suitable for required specification analog circuit topology using deep learning," in *2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2018, pp. 131–134.

[7] M. Rotman and L. Wolf, "Electric analog circuit design with hypernetworks and a differential simulator," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4157–4161.

[8] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2001.

[9] N. Horta, "Analogue and mixed-signal systems topologies exploration using symbolic methods," *Analog Integrated Circuits and Signal Processing*, p. 161–176, 2002.

[10] N. Jangkrajarng, S. Bhattacharya, R. Hartono, and C.-J. Shi, "Iprail—intellectual property reuse-based analog ic layout automation," *Integration, the VLSI Journal*, 2003, analog and Mixed-signal IC Design and Design Methodologies.

[11] R. Rutenbar, "Analog design automation: Where are we? where are we going?" in *Proc. CICC*, 1993.

[12] W. Daems, G. Gielen, and W. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE TCAD*, 2003.

[13] M. d. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a cmos op-amp via geometric programming," *IEEE TCAD*, 2001.

[14] Y. Wang, M. Orshansky, and C. Caramanis, "Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization," in *Proc. DAC*, 2014.

[15] W. Nye, D. Riley, A. Sangiovanni-Vincentelli, and A. Tits, "Delight.spice: an optimization-based system for the design of integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 4, pp. 501–519, 1988.

[16] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integr. VLSI J.*, vol. 42, no. 2, p. 137–148, Feb. 2009. [Online]. Available: https://doi.org/10.1016/j.vlsi.2008.04.003

[17] J. Koza, F. Bennett, D. Andre, M. Keane, and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 109–128, 1997.

[18] G. Alpaydin, S. Balkir, and G. Dundar, "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 240–252, 2003.

[19] R. A. Vural and T. Yildirim, "Swarm intelligence based sizing methodology for cmos operational amplifier," in *2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2011, pp. 525–528.

[20] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An efficient bayesian optimization approach for automated optimization of analog circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 6, pp. 1954–1967, 2017.

[21] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design," in *International conference on machine learning*. PMLR, 2018, pp. 3306–3314.

[22] S. Zhang, F. Yang, D. Zhou, and X. Zeng, "An efficient asynchronous batch bayesian optimization approach for analog circuit synthesis," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[23] S. Zhang, W. Lyu, F. Yang, C. Yan, D. Zhou, X. Zeng, and X. Hu, "An efficient multi-fidelity bayesian optimization approach for analog circuit synthesis," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.

[24] S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, "An efficient batch constrained bayesian optimization approach for analog circuit synthesis via multi-objective acquisition ensemble," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[25] B. He, S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, "An efficient bayesian optimization approach for analog circuit synthesis via sparse gaussian process modeling," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 67–72.

[26] A. Budak, M. Gandara, W. Shi, D. Pan, N. Sun, and B. Liu, "An efficient analog circuit sizing method based on machine learning assisted global optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2021.

[27] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H. Lee, and S. Han, "Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in *Proc. DAC*, 2020.

[28] K. Settaluri, A. Haj-Ali, Q. Huang, K. Hakhamaneshi, and B. Nikolić, "Autockt: Deep reinforcement learning of analog circuit designs," in *Proc. DATE*, 2020.

[29] A. F. Budak, P. Bhansali, B. Liu, N. Sun, D. Z. Pan, and C. V. Kashyap, "DNN-Opt an rl inspired optimization for analog circuit sizing using deep neural networks," in *Proceedings of the 58th ACM/EDAC/IEEE Design Automation Conference*, ser. DAC '21, 2021.

[30] R. Acar Vural and T. Yildirim, "Analog circuit sizing via swarm intelligence," *AEU - International Journal of Electronics and Communications*, 2012.

[31] B. Liu, G. Gielen, and F. V. Fernndez, *Automated Design of Analog and High-frequency Circuits: A Computational Intelligence Approach*. Springer, 2013.

[32] E. Afacan and G. Dundar, "A mixed domain sizing approach for rf circuit synthesis," 04 2016.

[33] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[34] A. Jain, J. Mao, and K. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[35] B. Liu, D. Zhao, P. Reynaert, and G. Gielen, "Synthesis of integrated passive components for high-frequency rf ics based on evolutionary computation and machine learning techniques," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30(10), pp. 1458 – 1468, 11 2011.

[36] B. Liu, D. Zhao, P. Reynaert, and G. G. E. Gielen, "Gaspad: A general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm," *IEEE TCAD*, Feb 2014.

[37] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proc. NIPS*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, 2012.

[38] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Multi-objective bayesian optimization for analog/rf circuit synthesis," in *Proc. DAC*.

[39] S. Zhang, W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Bayesian optimization approach for analog circuit synthesis using neural network," in *Proc. DATE*, 2019, pp. 1463–1468.

[40] G. İslamoğlu, T. O. Çakici, E. Afacan, and G. Dündar, "Artificial neural network assisted analog ic sizing tool," in *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2019, pp. 9–12.

[41] K. Hakhamaneshi, N. Werblun, P. Abbeel, and V. Stojanović, "Bagnet: Berkeley analog generator with layout optimizer boosted with deep neural networks," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019, pp. 1–8.

[42] Y. Li, Y. Wang, Y. Li, R. Zhou, and Z. Lin, "An artificial neural network assisted optimization system for analog design space exploration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2640–2653, 2020.

[43] N. Lourenço, J. Rosa, R. Martins, H. Aidos, A. Canelas, R. Póvoa, and N. Horta, "On the exploration of promising analog ic designs via artificial neural networks," in *2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2018, pp. 133–136.

[44] H. Chen, M. Liu, X. Tang, K. Zhu, N. Sun, and D. Z. Pan, "Challenges and opportunities toward fully automated analog layout design," *Journal of Semiconductors*, vol. 41, no. 11, p. 111407, nov 2020. [Online]. Available: https://doi.org/10.1088/1674-4926/41/11/111407

[45] J. Rijmenants, J. B. Litsios, T. R. Schwarz, and M. G. R. Degrauwe, "ILAC: an automated layout tool for analog cmos circuits," *IEEE Journal Solid-State Circuits*, vol. 24, no. 2, pp. 417–425, 1989.

[46] D. J. Chen, J. . Lee, and B. J. Sheu, "Slam: a smart analog module layout generator for mixed analog-digital vlsi design," in *Proc. ICCD*, 1989, pp. 24–27.

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

[48] S. Han, S. Jeong, C. Kim, H.-J. Park, and B. Kim, "GUI-Enhanced layout generation of ffe sst txs for fast high-speed serial link design," in *Proc. DAC*, 2020.

[49] D. Stefanovic, M. Kayal, M. Pastre, and V. B. Litovski, "Procedural analog design (pad) tool," in *Proc. ISQED*, 2003, pp. 313–318.

[50] R. Castro-Lopez, O. Guerra, E. Roca, and F. V. Fernandez, "An integrated layout-synthesis approach for analog ics," *IEEE TCAD*, vol. 27, no. 7, pp. 1179–1189, 2008.

[51] M. Ding, P. Harpe, G. Chen, B. Busze, Y. Liu, C. Bachmann, K. Philips, and A. van Roermund, "A hybrid design automation tool for sar adcs in iot," *IEEE TVLSI*, vol. 26, no. 12, pp. 2853–2862, 2018.

[52] C. Wulff and T. Ytterdal, "A Compiled 9-bit 20-MS/s 3.5-fJ/conv. step SAR ADC in 28-nm FDSOI for Bluetooth Low Energy Receivers," *IEEE Journal Solid-State Circuits*, vol. 52, no. 7, pp. 1915–1926, 2017.

[53] R. Martins, N. Lourenço, F. Passos, R. Póvoa, A. Canelas, E. Roca, R. Castro-López, J. Sieiro, F. V. Fernandez, and N. Horta, "Two-step rf ic block synthesis with preoptimized inductors and full layout generation in-the-loop," *IEEE TCAD*, vol. 38, no. 6, pp. 989–1002, 2019.

[54] J. Crossley, A. Puggelli, H. . Le, B. Yang, R. Nancollas, K. Jung, L. Kong, N. Narevsky, Y. Lu, N. Sutardja, E. J. An, A. L. Sangiovanni-Vincentelli, and E. Alon, "BAG: A designer-oriented integrated framework for the development of ams circuit generators," in *Proc. ICCAD*, 2013, pp. 74–81.

[55] E. Chang, J. Han, W. Bae, Z. Wang, N. Narevsky, B. NikoliC, and E. Alon, "BAG2: A process-portable framework for generator-based ams circuit design," in *Proc. CICC*, 2018, pp. 1–8.

[56] J. Han, E. Chang, S. Bailey, Z. Wang, W. Bae, A. Wang, N. Narevsky, A. Whitcombe, P. Lu, B. Nikolić, and E. Alon, "A generated 7gs/s 8b time-interleaved sar adc with 38.2db sndr at nyquist in 16nm cmos finfet," in *Proc. CICC*, 2019.

[57] E. Chang, N. Narevsky, J. Han, and E. Alon, "An automated serdes frontend generator verified with a 16-nm instance achieving 15 gb/s at 1.96 pj/bit," *IEEE Solid-State Circuits Letters*, vol. 1, no. 12, pp. 245–248, 2018.

[58] N. Mehta, S. Lin, B. Yin, S. Moazeni, and V. Stojanović, "A laser-forwarded coherent transceiver in 45-nm soi cmos using monolithic microring resonators," *IEEE Journal Solid-State Circuits*, vol. 55, no. 4, pp. 1096–1107, 2020.

[59] K. Kunal, M. Madhusudan, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, and S. S. Sapatnekar, "ALIGN: Open-source analog layout automation from the ground up," in *Proc. DAC*, 2019, pp. 1–4.

[60] B. Xu, K. Zhu, M. Liu, Y. Lin, S. Li, X. Tang, N. Sun, and D. Z. Pan, "MAGICAL: Toward fully automated analog ic layout leveraging human and machine intelligence: Invited paper," in *Proc. ICCAD*, 2019, pp. 1–8.

[61] H. Chen, M. Liu, X. Tang, K. Zhu, A. Mukherjee, S. Nan, and D. Z. Pan, "Magical 1.0: An open-source fully-automated ams layout synthesis framework verified with a 40-nm 1 gs/s $\delta\sigma$ adc,"," in *Proc. CICC*, 2021.

[62] H. Chen, K. Zhu, M. Liu, X. Tang, N. Sun, and D. Z. Pan, "Universal symmetry constraint extraction for analog and mixed-signal circuits with graph neural networks," in *Proc. DAC*, 2021, pp. 1–6.

[63] I.-P. Wu, H.-C. Ou, and Y.-W. Chang, "QB-trees: Towards an optimal topological representation and its applications to analog layout designs," in *Proc. DAC*, 2016, pp. 1–6.

[64] M. Liu, W. Li, K. Zhu, B. Xu, Y. Lin, L. Shen, X. Tang, N. Sun, and D. Z. Pan, "S3DET: Detecting system symmetry constraints for analog circuits with graph similarity," in *Proc. ASPDAC*, 2020, pp. 193–198.

[65] K. Zhu, H. Chen, M. Liu, and D. Z. Pan, "Automating analog constraint extraction: from heuristics to learning," in *Proc. ASPDAC*, 2022.

[66] A. Hastings and R. A. Hastings, *The Art of Analog Layout*, 1st ed. USA: Prentice Hall, 2001.

[67] J. Rijmenants, J. B. Litsios, T. R. Schwarz, and M. G. R. Degrauwe, "Ilac: an automated layout tool for analog cmos circuits," *IEEE Journal Solid-State Circuits*, vol. 24, no. 2, pp. 417–425, 1989.

[68] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, and L. R. Carley, "Koan/anagram ii: New tools for device-level analog placement and routing," *IEEE Journal Solid-State Circuits*, vol. 26, no. 3, pp. 330–342, 1991.

[69] F. Balasa and K. Lampaert, "Module placement for analog layout using the sequence-pair representation," in *Proc. DAC*, 1999, p. 274–279.

[70] Y. Pang, F. Balasa, K. Lampaert, and C.-K. Cheng, "Block placement with symmetry constraints based on the o-tree non-slicing representation," in *Proc. DAC*, 2000, p. 464–467.

[71] F. Balasa, S. C. Maruvada, and K. Krishnamoorthy, "Efficient solution space exploration based on segment trees in analog placement with symmetry constraints," in *Proc. ICCAD*, 2002, p. 497–502.

[72] ——, "Using red-black interval trees in device-level analog placement with symmetry constraints," in *Proc. ASPDAC*, 2003, p. 777–782.

[73] J.-M. Lin, G.-M. Wu, Y.-W. Chang, and J.-H. Chuang, "Placement with symmetry constraints for analog layout design using tcg-s," in *Proc. ASPDAC*, 2005, p. 1135–1137.

[74] F. Balasa, S. C. Maruvada, and K. Krishnamoorthy, "On the exploration of the solution space in analog placement with symmetry constraints," *IEEE TCAD*, vol. 23, no. 2, p. 177–191, Nov. 2006.

[75] D. Long, X. Hong, and S. Dong, "Signal-path driven partition and placement for analog circuit," in *Proc. ASPDAC*, 2006.

[76] J. Liu, S. Dong, Y. Ma, D. Long, and X. Hong, "Thermal-driven symmetry constraint for analog layout with cbl representation," in *Proc. ASPDAC*, 2007, p. 191–196.

[77] P.-H. Lin and S.-C. Lin, "Analog placement based on hierarchical module clustering," in *Proc. DAC*, 2008, p. 50–55.

[78] M. Strasser, M. Eick, H. Gräb, U. Schlichtmann, and F. M. Johannes, "Deterministic analog circuit placement using hierarchically bounded enumeration and enhanced shape functions," in *Proc. ICCAD*, 2008, p. 306–313.

[79] P.-H. Lin, Y.-W. Chang, and S.-C. Lin, "Analog placement based on symmetry-island formulation," *IEEE TCAD*, vol. 28, no. 6, pp. 791–804, 2009.

[80] L. Xiao and E. F. Y. Young, "Analog placement with common centroid and 1-d symmetry constraints," in *Proc. ASPDAC*, 2009, p. 353–360.

[81] P.-H. Lin, H. Zhang, M. D. F. Wong, and Y.-W. Chang, "Thermal-driven analog placement considering device matching," in *Proc. DAC*, 2009, p. 593–598.

[82] S. Nakatake, M. Kawakita, T. Ito, M. Kojima, M. Kojima, K. Izumi, and T. Habasaki, "Regularity-oriented analog placement with diffusion sharing and well island generation," in *Proc. ASPDAC*, 2010, p. 305–311.

[83] C.-W. Lin, J.-M. Lin, C.-P. Huang, and S.-J. Chang, "Performance-driven analog placement considering boundary constraint," in *Proc. DAC*, 2010, p. 292–297.

[84] Q. Ma, L. Xiao, Y.-C. Tam, and E. F. Young, "Simultaneous handling of symmetry, common centroid, and general placement constraints," *IEEE TCAD*, p. 85–95, Jan. 2011.

[85] H.-F. Tsao, P.-Y. Chou, S.-L. Huang, Y.-W. Chang, M. P.-H. Lin, D.-P. Chen, and D. Liu, "A corner stitching compliant b*-tree representation and its applications to analog placement," in *Proc. ICCAD*, 2011, p. 507–511.

[86] P.-Y. Chou, H.-C. Ou, and Y.-W. Chang, "Heterogeneous b*-trees for analog placement with symmetry and regularity considerations," in *Proc. ICCAD*, 2011, p. 512–516.

[87] C.-W. Lin, C.-C. Lu, J.-M. Lin, and S.-J. Chang, "Routability-driven placement algorithm for analog integrated circuits," in *Proc. ISPD*, 2012, p. 71–78.

[88] P.-H. Wu, M. P.-H. Lin, Y.-R. Chen, B.-S. Chou, T.-C. Chen, T.-Y. Ho, and B.-D. Liu, "Performance-driven analog placement considering monotonic current paths," in *Proc. ICCAD*, 2012.

[89] H.-C. C. Chien, H.-C. Ou, T.-C. Chen, T.-Y. Kuan, and Y.-W. Chang, "Double patterning lithography-aware analog placement," in *Proc. DAC*, 2013.

[90] H.-C. Ou, H.-C. C. Chien, and Y.-W. Chang, "Simultaneous analog placement and routing with current flow and current density considerations," in *Proc. DAC*, 2013.

[91] P.-H. Wu, M. P.-H. Lin, T.-C. Chen, C.-F. Yeh, T.-Y. Ho, and B.-D. Liu, "Exploring feasibilities of symmetry islands and monotonic current paths in slicing trees for analog placement," *IEEE TCAD*, vol. 33, no. 6, pp. 879–892, 2014.

[92] I.-P. Wu, H.-C. Ou, and Y.-W. Chang, "Qb-trees: Towards an optimal topological representation and its applications to analog layout designs," in *Proc. DAC*, 2016.

[93] H.-C. Ou, K.-H. Tseng, J.-Y. Liu, I.-P. Wu, and Y.-W. Chang, "Layout-dependent effects-aware analytical analog placement," *IEEE TCAD*, vol. 35, no. 8, pp. 1243–1254, 2016.

[94] B. Xu, S. Li, X. Xu, N. Sun, and D. Z. Pan, "Hierarchical and analytical placement techniques for high-performance analog circuits," in *Proc. ISPD*, 2017.

[95] Y.-S. Lu, Y.-H. Chang, and Y.-W. Chang, "WB-Trees: A meshed tree representation for finfet analog layout designs," in *Proc. DAC*, 2018.

[96] A. Patyal, P.-C. Pan, A. K. A, H.-M. Chen, H.-Y. Chi, and C.-N. Liu, "Analog placement with current flow and symmetry constraints using PCP-SP," in *Proc. DAC*, 2018.

[97] B. Xu, B. Basaran, M. Su, and D. Z. Pan, "Analog placement constraint extraction and exploration with the application to layout retargeting," in *Proc. ISPD*, 2018.

[98] B. Xu, S. Li, C.-W. Pui, D. Liu, L. Shen, Y. Lin, N. Sun, and D. Z. Pan, "Device layer-aware analytical placement for analog circuits," in *Proc. ISPD*, 2019.

[99] K. Zhu, H. Chen, M. Liu, X. Tang, N. Sun, and D. Z. Pan, "Effective analog/mixed-signal circuit placement considering system signal flow," in *Proc. ICCAD*, 2020.

[100] K. Zhu, H. Chen, M. Liu, X. Tang, W. Shi, N. Sun, and D. Z. Pan, "Generative-adversarial-network-guided well-aware placement for analog circuits," in *Proc. ASPDAC*, 2022.

[101] N. Jangkrajarng, L. Zhang, S. Bhattacharya, N. Kohagen, and C.-J. R. Shi, "Template-based parasitic-aware optimization and retargeting of analog and rf integrated circuit layouts," in *Proc. ICCAD*, 2006, p. 342–348.

[102] U. Choudhury and A. Sangiovanni-Vincentelli, "Constraint-based channel routing for analog and mixed analog/digital circuits," *IEEE TCAD*, vol. 12, no. 4, pp. 497–510, 1993.

[103] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, and L. R. Carley, "Koan/anagram ii: new tools for device-level analog placement and routing," *IEEE Journal Solid-State Circuits*, vol. 26, no. 3, pp. 330–342, March 1991.

[104] P. Lin, H. Yu, T. Tsai, and S. Lin, "A matching-based placement and routing system for analog design," in *Proc. VLSI-DAT*, April 2007, pp. 1–4.

[105] H. Ou, H. C. Chien, and Y. Chang, "Non-uniform multilevel analog routing with matching constraints," in *Proc. DAC*, June 2012, pp. 549–554.

[106] P. Pan, H. Chen, Y. Cheng, J. Liu, and W. Hu, "Configurable analog routing methodology via technology and design constraint unification," in *Proc. ICCAD*, Nov 2012, pp. 620–626.

[107] L. Xiao, E. F. Y. Young, X. He, and K. P. Pun, "Practical placement and routing techniques for analog circuit designs," in *Proc. ICCAD*, 2010, pp. 675–679.

[108] C. Wu, H. Graeb, and J. Hu, "A pre-search assisted ilp approach to analog integrated circuit routing," in *Proc. ICCD*, Oct 2015, pp. 244–250.

[109] H. Chi, H. Tseng, C. J. Liu, and H. Chen, "Performance-preserved analog routing methodology via wire load reduction," in *Proc. ASPDAC*, Jan 2018, pp. 482–487.

[110] Q. Gao, Y. Shen, Y. Cai, and H. Yao, "Analog circuit shielding routing algorithm based on net classification," in *Proc. ISLPED*, Aug 2010, pp. 123–128.

[111] M. M. Ozdal and R. F. Hentschke, "Exact route matching algorithms for analog and mixed signal integrated circuits," in *Proc. DAC*, Nov 2009, pp. 231–238.

[112] ——, "Maze routing algorithms with exact matching constraints for analog and mixed signal designs," in *Proc. ICCAD*, Nov 2012, pp. 130–136.

[113] H. Ou, H. Chang Chien, and Y. Chang, "Simultaneous analog placement and routing with current flow and current density considerations," in *Proc. DAC*, May 2013, pp. 1–6.

[114] J.-W. Lin, T.-Y. Ho, and I. H.-R. Jiang, "Reliability-driven power/ground routing for analog ics," in *ACM TODAES*, vol. 17, no. 1, 2012, pp. 6:1–6:26.

[115] R. Martins, N. Lourenço, A. Canelas, and N. Horta, "Electromigration-aware and ir-drop avoidance routing in analog multiport terminal structures," in *Proc. DATE*, March 2014, pp. 1–6.

[116] H. Chen, K. Zhu, M. Liu, X. Tang, N. Sun, and D. Z. Pan, "Toward silicon-proven detailed routing for analog and mixed-signal circuits," in *Proc. ICCAD*, 2020.

[117] H. Chen, M. Liu, B. Xu, K. Zhu, X. Tang, S. Li, Y. Lin, N. Sun, and D. Z. Pan, "Magical: An open-source fully automated analog ic layout system from netlist to gdsii," *IEEE Design & Test*, 2020.

[118] B. Xu, Y. Lin, X. Tang, S. Li, L. Shen, N. Sun, and D. Z. Pan, "WellGAN: Generative-adversarial-network-guided well generation for analog/mixed-signal circuit layout," in *Proc. DAC*, 2019, pp. 1–6.

[119] K. Zhu, M. Liu, Y. Lin, B. Xu, S. Li, X. Tang, N. Sun, and D. Z. Pan, "GeniusRoute: A new analog routing paradigm using generative neural network guidance," in *Proc. ICCAD*, 2019, pp. 1–8.

[120] M. Liu, K. Zhu, J. Gu, L. Shen, X. Tang, N. Sun, and D. Z. Pan, "Towards decrypting the art of analog layout: Placement quality prediction via transfer learning," in *Proc. DATE*, 2020, pp. 1–6.

[121] Y. Li, Y. Lin, M. Madhusudan, A. Sharma, W. Xu, S. Sapatnekar, R. Harjani, and J. Hu, "A customized graph neural network model for guiding analog ic placement," in *Proc. ICCAD*, 2020.

[122] R. Reis, Y. Cao, and G. Wirth, *Circuit design for reliability*. Springer, 2015.

[123] X. Li and L. Pileggi, *Statistical Performance Modeling and Optimization*. Now Publishers, 2007.

[124] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *2006 43rd ACM/IEEE Design Automation Conference*. IEEE, 2006, pp. 69–72.

[125] S. Sun, X. Li, H. Liu, K. Luo, and B. Gu, "Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1096–1109, 2015.