Intelligent and Interactive Analog Layout Design Automation

Yibo Lin[1,3⋆], Xiaohan Gao[2,1], Haoyi Zhang[1], Runsheng Wang[1,3], Ru Huang[1,3]
[1]School of Integrated Circuits, Peking University [2]School of Computer Science, Peking University
[3]Beijing Advanced Innovation Center for Integrated Circuits, Beijing, China

***Abstract*—Automating analog layout design has long been considered a challenging problem. The literature has been exploring fully automated analog layout tools, while designers still adopt manual methodologies in practice, due to the performance gap between generated layouts and manually crafted ones. Although fully automated tools cannot generate flawless solutions yet, we can enhance the design practice by incorporating designers' expertise with intelligent and efficient automation process. In this paper, we discuss two potential directions to leverage designers' expertise, i.e., by utilizing machine intelligence to learn from existing manual layouts, and allowing designers to efficiently edit layouts with real-time interaction. We survey the recent work dedicated to these two directions and hope to inspire more studies.**

I. Introduction

Analog layout design has suffered from a low level of automation for decades. Despite the recent efforts in fully automated layout generation tools [1], [2], the performance gap between generated layouts and manually crafted layouts still exists. As a result, designers still prefer manual design methodologies in practice. To enhance the performance of layout automation tools, the literature starts exploring ways to incorporate machine intelligence and human interaction during the design process.

Leveraging machine intelligence like deep learning or deep neural networks (DNN) can help extract human knowledge from existing manual analog layouts. For example, analog layout designs usually require versatile design constraints, such as symmetry, matching, etc. Annotating all constraints for each design and feeding to automation tools can be a tedious work. An analog circuit can be naturally abstracted to a hypergraph, and thus graph neural networks (GNNs) are suitable for learning the annotation tasks. Several researches have unleashed the power of GNNs on symmetry constraint annotation [3], [4], [5], [6]. Other studies further try to explore the possibility of directly generating partial or entire layouts with generative models [7], [8].

Human interaction like interactive layout editing is another promising way to take advantage of designers' expertise. Unlike the deep-learning-based layout automation approaches making use of existing layouts, interactive layout editing embraces human involvement when developing a layout with automation tools [9], [10]. As an analog design could have many specifications and different design styles, users may not always be satisfied with generated layouts. Interactive layout editing overcomes the drawback of fully automated layout generation that does not allow any user engagement. Designers can edit the generated layout to approach their expectation, but get freed from drawing an initial layout and manually resolving design rule violations. In other words, interactive layout editing promotes efficient use of human knowledge and automation tools.

⋆Corresponding author: yibolin@pku.edu.cn

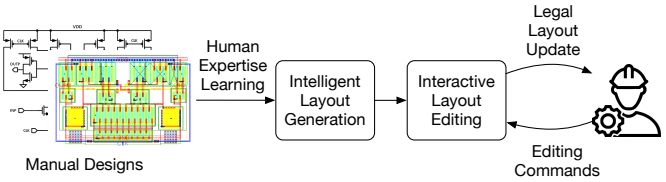


Fig. 1: Overall flow for intelligent and interactive analog layout automation.

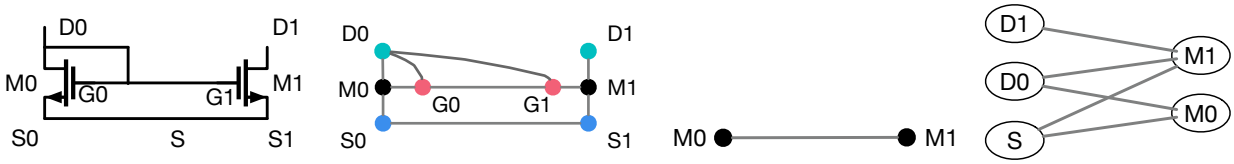
By surveying the literature, we observe that machine intelligence can be helpful to learn human expertise and guide initial layout generation, while interactive layout editing can close the gap between generated layouts and human expectation. We summarize the overall flow of such an intelligent and interactive desgin methodology in Figure 1 and discuss the related progress in this paper. The following sections are organized as follows: Section II surveys the application of deep learning for analog layout generation, Section III investigates the studies in interactive analog layout editing, and Section IV concludes the paper.

II. Intelligent Layout Generation

Rapid development of deep learning stimulates the research for analog layout generation leveraging machine intelligence. The studies can be mainly divided into three aspects: 1) learn to annotate design constraints, such as symmetry constraints [5], [3], [4], [6]; 2) guide analytical layout generation algorithms [11], [12]; 3) generate partial or entire layouts directly [7], [8].

A. Symmetry Constraint Annotation

Symmetry constraint is widely adopted in analog design to guarantee matching and design performance. Analog layout automation tools [13], [2] take a circuit netlist as input and generate symmetry constraints for downstream layout



(a) A circuit example (b) Graph representation [3] (c) Graph representation [4] (d) Graph representation [5]

Fig. 2: Different graph representations.

generation stages, like device placement and routing. Early studies for symmetry annotation include circuit performance analysis [14], graph matching [15], pattern matching [16], and graph similarity [17]. These approaches either suffer from scalability issues of circuit sizes [14] or flexibility issues requiring comprehensive pattern libraries [16] and empirical tuning of matching regions [15], [17].

Recent advancements in graph neural networks (GNNs) open up a new direction for symmetry constraint annotation, as an analog circuit is essentially a graph. We can leverage GNNs to learn the annotation task and predict symmetry constraints on unseen circuits. There are different ways to abstract an analog circuit to a graph [3], [4], [5], [6], as shown in Figure 2. Gao et al. [3] map devices and pins to graph nodes and represents the hypergraph of the circuit in Figure 2(a) with cliques. Chen et al. [4] adopt a similar representation but consider only devices. Kunal et al. [5] adopt a bipartite graph representation with all the nets, devices and pins as graph nodes. These different graph representations help embed structural features for symmetry annotation. With graph abstraction, symmetry annotation can be converted into a binary classification task of each node pair. By leveraging GNN to encode node embedding, Gao et al. [3] propose a GraphSage-based model [18] to determine whether a device-level symmetry exists between a piar of devices. Chen et al. [4] further encode graph embedding to tackle the system-level symmetry between sub-circuits. Kunal et al. [5] adopts graph convolutional networks to identify functionalities of hierarchical circuit blocks and derive the symmetry constraints from a pre-recorded library.

B. Guidance for Placement & Routing

Analog layout generation typically includes placement and routing stages, which involve to solve optimization problems. Conventional analog placement and routing algorithms adopt analytical optimization [19], [20], [21] or heuristic searching approaches like simulated annealing [22]. These methods require explicit mathematical formulation, which may not be able to leverage the design expertise from existing layouts.

Li et al. [11] propose to guide analog placement with graph neural networks. As post-layout simulation is time-consuming and there is no routing information at the placement stage, they utilize graph neural networks to predict the impact of placement on post-layout simulation performance. The proposed performance prediction model is integrated into the objective of a simulated-annealing-based placer. Experimental results show that the placer has the potential to achieve comparable performance to manually crafted layouts.

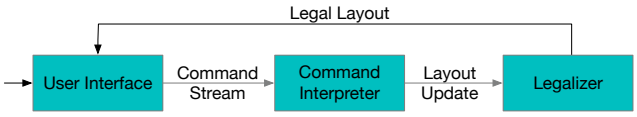


Fig. 3: The flow of interactive layout editing.

Zhu et al. [12] propose an analog routing paradigm enhanced by a generative model that provides guidance for the router to mimic the behavior of manual routing. The generative model is based on variational autoencoder [23], which learns the routing probability map from manual layouts. The probability map is then incorporated into the cost function of the A-star routing algorithm. They demonstrate significant improvements on specific performance metrics with such guidance.

C. Partial Layout Generation

In analog design, wells or doping areas are usually inserted after device placement. This is not an easy task, as the solutions generated by analytical algorithms are far from manual ones. Xu et al. [7] propose WellGAN, to leverage generative adversarial networks (GANs) for well generation to mimic the manually crafted well patterns. With a post legalization step to satisfy design rules and enclose all the target transistors, WellGAN can produce well solutions similar to manual ones.

Gusmão et al. [8] propose to directly generate analog placement results with a graph-to-sequence model, AGraph2Seq. AGraph2Seq formulates placement constraints, including proximity constraints, current-flow constraints, and symmetry constraints, as input, and sequentially outputs the position of each device. As a deep learning model may not generate legal placement solutions, AGraph2Seq still needs analytical approaches to satisfy all the design constraints.

III. Interactive Layout Editing

Interactive analog layout editing can serve as a bridge connecting manual engineering efforts and automated layout generation tools. The dilemma in analog layout design is that manually drawing a layout can be time consuming, while the layouts produced by fully automated tools are not satisfactory enough for complex design requirements. An eclectic methodology is to generate an initial layout with automation tools and involve designers for interactive layout editing, as shown in Figure 3.

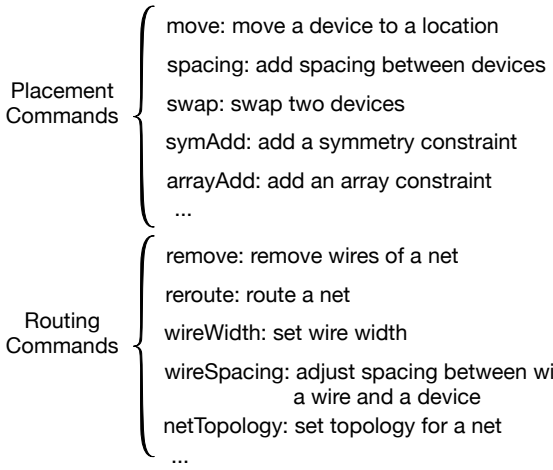


Fig. 4: Interactive layout editing commands.

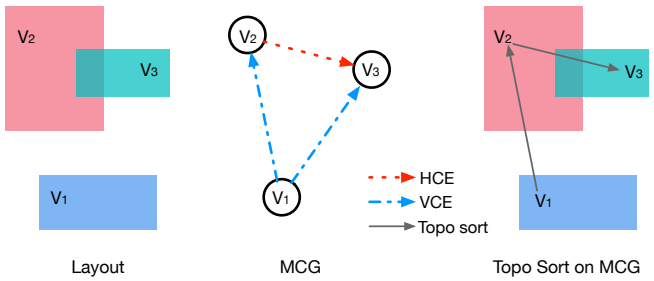


Fig. 5: Mixed constraint graph and the legalization algorithm.

A. Interactive Placement

Gao et al. [9] propose an interactive layout editing paradigm for analog placement. As shown in Figure 1, interactive placement starts with an initial placement solution from fully automated tools. A designer can view the layout through visualization and provides editing commands through a user interface. Then, the designer interacts with the layout editing tool and gets the desired layout in several edit-update-legalize-show iterations.

The authors propose an extensible command set for manipulating the placement layout, including {`move`, `spacing`, `swap`, `arrayAdd`, `symAdd`}. Commands `move`, `spacing`, and `swap` define fine-grained operations on device locations. Commands `arrayAdd` and `symAdd` introduce high-level constraint-related operations. Designers can edit the placement solution with these commands and view the updated legal results at real-time.

In order to meet the requirement of real-time interaction, a new graph representation that combines the horizontal constraint graph (HCG) and vertical constraint graph (VCG) for analog placement is proposed, i.e., mixed constraint graph (MCG). As shown in Figure 5, a MCG consists of two kinds of constraint edges, horizontal constraint edge (HCE), and vertical constraint edge (VCE). HCE means that some device is right to another device, like device V_2 and device V_3. VCE is similar, which stands for the up-down relationship. MCG possesses a helpful property with its topological sort:

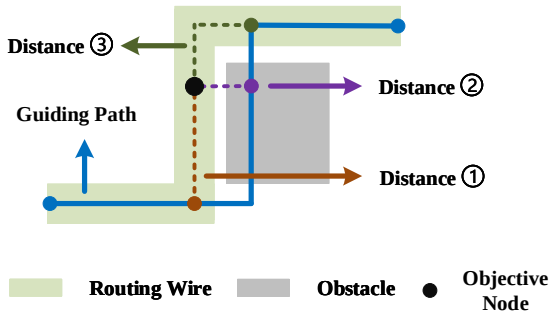


Fig. 6: Example of routing with topology guidance.

given a topological sort of a MCG, moving any device to its top left has no impact on the devices before that device in the topological order. "No impact" means that it will not cause new overlaps or introduce new illegal placement parts. Then, the authors propose a linear-time legalization algorithm that traverses the MCG once and eliminates the illegal placement part by simply moving the devices to up right. Symmetry constraints can also be handled by splitting the MCG according to the symmetry axis. The right figure of Figure 5 shows a topological sort of the MCG. The algorithm traverses the devices in the order of $\{V_1, V_2, V_3\}$ and moves V_3 to its up right to eliminate the overlap between itself and V_2, which is a simple run of the legalization algorithm.

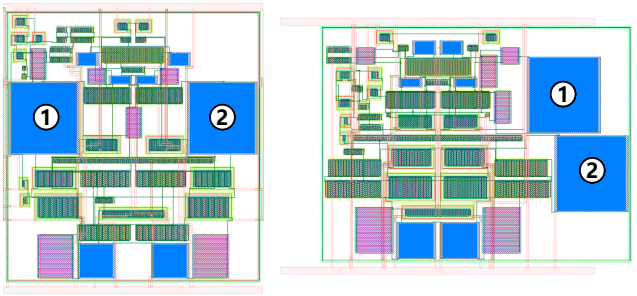
B. Interactive Routing

Routing is the most tedious and time-consuming procedure in the analog layout design flow due to sophisticated constraints, especially when designers have to manually finish all the wiring. Interactive routing aims at incorporating designers' expertise to close the performance gap [10]. Given the initial routing solution generated by automation tools, designers are able to adjust the routing topology with commands and get a legal routing solution without DRC violations instantly.

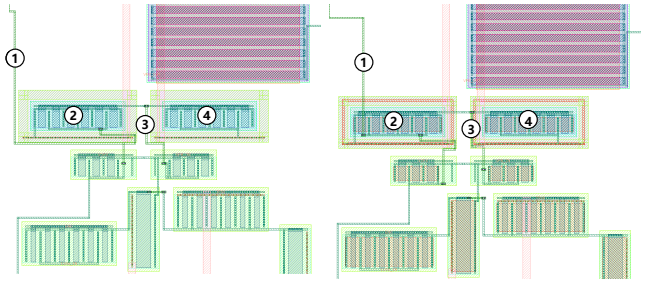
The editing commands for routing are shown in Figure 4, including basic commands like removing and rerouting a single net, setting particular wire width, as well as superior commands like adjusting the spacing between two designated wires and specifying rough routing topologies for nets. Take the `netTopology` command as an example. It allows designers to set topology guidance for specific nets. As demonstrated in Figure 6, we minimize the distance between the objective nodes and the guiding path when performing the A-star-based routing algorithm. We consider the Manhattan distance between the objective node and each segment of the guiding path as the distance metric. In this way, the routing algorithm will generate a routing path close to the guiding path and meanwhile avoid obstacles.

C. Results for Demonstration

Figure 7 demonstrates the results of interactive analog layout editing on two real-world circuits, OTA and LDO. We perform interactive placement on OTA by adding a symmtery



(a) Entire layout of OTA



(b) Partitial layout of LDO

Fig. 7: An example of interactive analog layout flow with OTA and LDO.

constraint on devices ① and ②. We also perform interactive routing on LDO and get a legalized routing topology by removing wires ① & ③ from electrically sensitive regions ② & ④. The post-layout simulation performance of the two circuits is significantly improved after interactive layout editing [10].

IV. Conclusions

With growing challenges in analog layout design, analog layout automation becomes a promising direction to speedup design cycles. We point out that existing layout automation tools can be enhanced with machine intelligence and human interaction. In this paper, we present the recent techniques on deep learning for analog layout generation and an interactive paradigm for analog layout editing. We hope this paper will inspire future work on intelligent and interactive analog layout automation.

Acknowledgement

This project is supported in part by the National Natural Science Foundation of China (No. 62141404), Hisilicon, and the 111 Project (B18001).

References

[1] T. Dhar, K. Kunal, Y. Li, M. Madhusudan, J. Poojary, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, P. Mukherjee, S. Yaldiz, and S. S. Sapatnekar, "Align: A system for automating analog layout," *IEEE MDAT*, vol. 38, no. 2, pp. 8–18, 2021.

[2] 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," in *Proc. ICCAD*. IEEE, 2019, pp. 1–8.

[3] X. Gao, C. Deng, M. Liu, Z. Zhang, D. Z. Pan, and Y. Lin, "Layout symmetry annotation for analog circuits with graph neural networks," in *Proc. ASPDAC*, Tokyo, Japan, 2021, p. 152–157.

[4] 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. 1243–1248.

[5] K. Kunal, T. Dhar, M. Madhusudan, J. Poojary, A. Sharma, W. Xu, S. M. Burns, J. Hu, R. Harjani, and S. S. Sapatnekar, "Gana: Graph convolutional network based automated netlist annotation for analog circuits," in *Proc. DATE*, 2020, pp. 55–60.

[6] K. Kunal, J. Poojary, T. Dhar, M. Madhusudan, R. Harjani, and S. S. Sapatnekar, "A general approach for identifying hierarchical symmetry constraints for analog circuit layout," in *Proc. ICCAD*, ser. Proc. ICCAD, Virtual Event, USA, 2020.

[7] 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*, Las Vegas, NV, USA, 2019.

[8] A. P. L. de Gusmão, N. C. Gomes Horta, N. C. Correia Lourenço, and R. M. Ferreira Martins, "Scalable and order invariant analog integrated circuit placement with attention-based graph-to-sequence deep models," *Expert Systems with Applications*, vol. 207, p. 117954, 2022.

[9] X. Gao, M. Liu, D. Z. Pan, and Y. Lin, "Interactive analog layout editing with instant placement legalization," in *Proc. DAC*, 2021, pp. 1249–1254.

[10] X. Gao, H. Zhang, M. Liu, L. Shen, D. Z. Pan, Y. Lin, R. Wang, and R. Huang, "Interactive analog layout editing with instant placement and routing legalization," *IEEE TCAD*, 2022.

[11] Y. Li, Y. Lin, M. Madhusudan, A. Sharma, W. Xu, S. S. Sapatnekar, R. Harjani, and J. Hu, "A customized graph neural network model for guiding analog ic placement," in *Proc. ICCAD*, 2020, pp. 1–9.

[12] 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.

[13] 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.

[14] E. Charbon, E. Malavasi, and A. Sangiovanni-Vincentelli, "Generalized constraint generation for analog circuit design," in *Proc. ICCAD*, 1993, pp. 408–414.

[15] M. Eick, M. Strasser, K. Lu, U. Schlichtmann, and H. E. Graeb, "Comprehensive generation of hierarchical placement rules for analog integrated circuits," *IEEE TCAD*, vol. 30, no. 2, pp. 180–193, 2011.

[16] P.-H. Wu, M. P.-H. Lin, and T.-Y. Ho, "Analog layout synthesis with knowledge mining," in *2015 European Conference on Circuit Theory and Design (ECCTD)*, 2015, pp. 1–4.

[17] 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*. IEEE, 2020, pp. 193–198.

[18] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NeurIPS*, Long Beach, California, USA, 2017, p. 1025–1035.

[19] 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*, San Francisco, CA, USA, 2019, p. 19–26.

[20] 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*. IEEE, 2020.

[21] 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, pp. 1–8.

[22] 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.

[23] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.