Semi-Automated Analog Placement based on Margin Tolerances

Eric Lao, Marie-Minerve Louerat, Jean-Paul Chaput

Semi-Automated Analog Placement based on Margin Tolerances

Eric Lao, Marie-Minerve Louërat, Jean-Paul Chaput
Sorbonne Universités, UPMC Univ Paris 06, CNRS, Laboratoire d'Informatique de Paris 6 (LIP6), Paris, France
Eric.Lao@lip6.fr, Marie-Minerve.Louerat@lip6.fr, Jean-Paul.Chaput@lip6.fr

Abstract— Digital circuit design is extensively assisted by modern automation tool unlike analog design which is still a manual task because of the complexity of the interactions between devices. This paper presents a semi-automated analog placement based on margin tolerances controlled by the designer by creating analog circuits organized in row similar to digital circuits structure. The results show the ability of our tool at generating multiple layouts respecting designer's constraints.

I. Introduction

Modern system-on-chip contains both digital and analog circuit. Even though most functions in an integrated circuit are implemented with digital circuitry, some functions rely on analog circuit and are the link with the external world. Despite the fact that the digital design flow has been well-defined over the last years, the analog design flow lacks a common methodology making it difficult to capture a design procedure.

An automatic placement tool should produce analog device-level layouts similar in density and performance to the high-quality manual layouts. To achieve this task, the capability to deal with layout constraints, in order to eliminate unwanted parasitics due to the process variations, is mandatory. [1] enumerates the most common placement constraints that are respected by modern analog placer. Among them, symmetries and symmetry-island [4] are the most used constraints. Other constraints like common-centroid [5], proximity and range are also often considered [6]. Taking into account current and signal path improves performance accuracy [7]. Devices can be placed depending on their thermal impact on the chip [8]. Regularity [9] and boundary constraints [10] enhance routability and suppress parasitics induced by extra bends of wires and via cost.

Recent research focused on using simulated annealing algorithm (SA) in combination with topological representations to respect these placement constraints. Topological representations encode the positioning relations between devices and SA optimizer alters their relative positions. The most popular representations used in analog placement over the last decades are Sequence-Pair [11], B*-Tree [12], Transitive Closure Graph [13], Ordered-Tree [14] and slicing floorplans [15] and they were coupled with some constraints to respect at the same time.

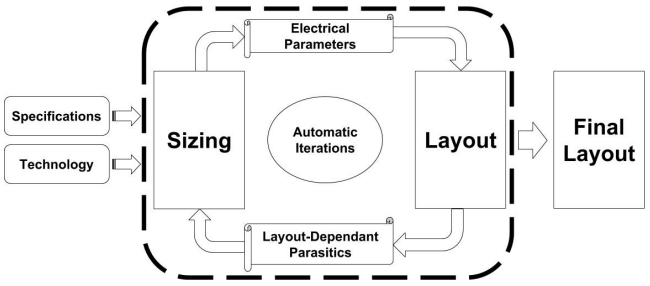

Fig. 1. Cairo Hurricane AMS Design Flow

Although most of the recent works focus on simulated annealing algorithm [1], we believe that giving more control to designers and using their interventions to set some constraints yield good analog placement results. In this paper, we introduce a semi-automatic analog placement approach guided by designers' preferences. This semi-automatic approach also helps designers to debug more efficiently and makes adjustments easier since they will control the overall relative placement of the circuit but at the same time, some tiresome and error-prone tasks are automated.

The remainder of this paper is organized as follows: Section II describes the analog/mixed-signal placement approach. Section III presents our methodology. Section IV explains the experimental results. Conclusion and future challenges are given in Section V.

II. Analog/Mixed-signal Design Approach

A. Design Flow

The Cairo Hurricane AMS (CHAMS) project [16], developed by the LIP6 Laboratory, proposes a complete flow which would be able to create a library of reconfigurable analog Intellectual Properties, to automatically generate and optimize the layout with little intervention from designers. The flow provides a reliable and efficient solution taking into account parasitic effect-aware layout generation with enough flexibility to adapt to different designers needs and concerns. Our layout generation tool supports any technology with the new nanometric layout dependent parasitic parameters.

The proposed CHAMS analog design (Fig.1) is a two

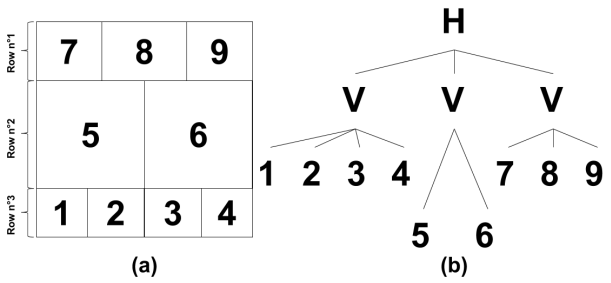

Fig. 2. Example of placement (a) and its slicing tree representation (b) where "H" stands for horizontal cut and "V" for vertical cut

way communication between the sizing and layout generation. The idea is that the sizing tool provides the electrical parameters of the transistor such as the width, length, number of fingers, etc... to the layout generation tool. The tool automatically generates the layout from a library of parameterized cells and sends back to the sizing engine the layout-dependent parasitic parameters such as the drain and source areas and perimeters, the stress effect parameters, etc... to re-evaluate the performance. This internal loop is repeated several times, with minimal designers intervention, until the target specifications are met.

This paper presents the placement approach of our layout generation tool. As mentioned in the previous paragraph, we emphasize that it is part of an internal loop which will be repeated several times. Therefore our placement algorithm does not provide an optimized analog layout in one run. It results from multiple adjustments (sizing and shape) performed by the loop shown in (Fig.1). Having these considerations in mind, we propose this placement approach because we believe it can provide a better control over the circuit placement and especially easier to adjust compared to simulated annealing approaches.

B. Placement in row

Digital and analog circuits have a dedicated area on a system-on-chip circuit so they can be independently designed within a specific space. Digital circuits are well-known for their regular row structure where standard cells are placed and routed accordingly to their netlist. In a similar way, we plan to organize the analog circuit layout in rows of devices and the analog circuit area should be placed and routed within its dedicated area.

It is common to design analog circuit in rows of devices where the height of each row of devices should be adjustable so it can match its dedicated area. Therefore, we choose the slicing tree representation for several reasons:

- Slicing trees are a natural choice since they are adequate to the row structure. Rows are represented by horizontal slices which will be divided into vertical slices determined by the area of each device.
- Unlike most of modern analog placement methods, our placement strategy is semi-automatic and will be guided by designers' preferences. Slicing trees are easy to handle and let designers choose the overall topology. That shows some advantages that will be explained in the following section.

III. Methodology

Modern analog automation tools are able to generate analog layout respecting various placement constraints and they mainly use simulated annealing approach. Nevertheless, such approach might not produce predictable and easy-to-adjust results. We believe that giving more control to designers will generate analog layout, easier to predict compared to optimization-based approach, but also easier to adjust in case modifications are required. Our analog/mixed-signal placer is semi-automatic: the device generation is automated and generated correct-by-construction and guided by designers' constraints over the circuit. The following subsections will describe the designers interventions.

A. Slicing Tree Construction

In order to organize devices in row, we use the slicing tree structure (Fig.2) which will be described the designers and not automatically generated by the tool. A slicing tree is a slicing floorplan that represents an area that has been divided multiple times either vertically or horizontally, forming a set of rectangular regions representing the place filled by each device. These slices are organized hierarchically so they form a graph where the hierarchical node are either horizontal or vertical cuts. Fig.2 shows an example of a slicing tree representing a circuit of 9 devices organized in three rows. We insist on the fact that the slicing tree is specified by designers who will also precise the analog constraints in the slices of the slicing tree. Unlike previous topology extraction approaches, our analog layout generation tool considers the designers' slicing tree as an input of the flow.

B. Devices Variations

Our placement approach consists in organizing devices in row, that is to say to have rows of devices with similar height, and the analog part's height would be a multiple of standard cell's heights from the digital circuit part. To obtain a row of devices, the height of each device needs to be similar and this is the reason why, we need to consider several possible aspect ratios for each device by varying the number of fingers like in [7] so we can find heights of devices that match a given height.

As [3] mentioned, changing the number of fingers of a device can considerably change the device property, especially the source/drain bulk capacitance. Nevertheless, our

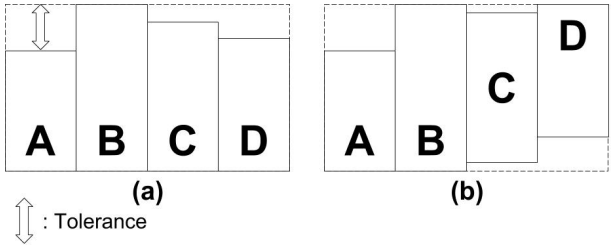

Fig. 3. Tolerance (a) and alignment (b) in slicing tree representation

placement phase is part of an internal loop (Fig.1), which is repeated until a result meets the required performances (Fig.1). Also, it is the task of designers to decide the number variations for every device which would limit the number of finger variations.

C. Margin Tolerances

We aim at creating rows and it is important to define what we consider as a row of devices since the devices will rarely have exactly the same height. We introduce a tolerance margin, which represents the difference between the smallest and the tallest device's height in a vertical node. Fig.3 (a) shows an example of devices organized in one row where the area A, B, C and D represents devices. Devices A and B show respectively the smallest and highest height inside the row. This difference of height is compared to the tolerance parameter to establish if this is considered as a row. We apply the same concept to the horizontal node but instead of using the height as a comparison, we use the width. It is up to designers to adjust this tolerance margin and it will impact on the number of accepted possibilities. Different margin tolerances can be considered at each hierarchical nodes. At the same time, having a small tolerance will reduce the waste of space induced by the slicing tree representation.

D. Placement Constraints

As said in section III-A, it is the task of designers to describe the slicing tree with the help of our tool. This will directly impact on the topology of the circuit. This means designers will have the total control over the relative relation between the blocks. Having designers building their own slicing tree also means that they will have control over placement constraints such as proximity, boundary, current/signal path and regularity constraints based on their knowledges and preferences.

Among the most common constraints, symmetries can be respected with the appropriate slicing tree organization. We also consider alignment constraints inside a slicing tree in horizontal or vertical node. In Fig.3 (b), we have an example of possible alignments: devices A and B are aligned to the bottom of the row, device C is centered and device D is aligned to the top of the row. In horizontal slices, devices can be aligned to the right, center and left of the row.

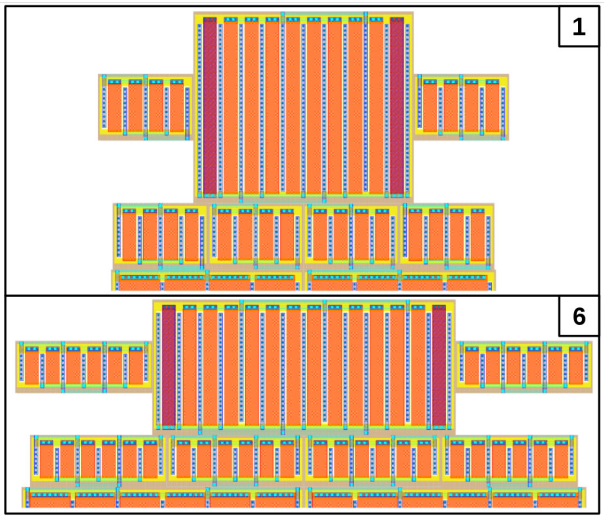

Fig. 4. Evolution of 2 rows from layout 1 and 6 of Table I for different global aspect ratios. These figures have the same scale.

E. Placement Choice

Similar to [17], once all the possible variations are set for each device, we evaluate the accepted variations at hierarchical nodes based on the margin tolerances. These accepted variations are propagated from the bottom of the slicing tree to the root. After this bottom-up propagation, designers process the different placements based on height, width or global ratio criteria. They can choose the most optimized placements according to the Pareto front curve like in [18], but they are free to choose any other possible placement that would eventually have more white space if the circuit affords more space.

IV. Results

Our tool was implemented in C++ programming language on a Intel(R) Core(TM) i5-4590S CPU @ 3.00 GHz workstation with 6 GB RAM. To illustrate the capability of our tool, we experiment it on a fully differential transconductor [19],

TABLE I
Some layout results of the fully differential transconductor

Layout	Area (μm^2)	Width (μm)	Height (μm)	Occupation (%)
1	4600	84	54	68
2	5116	91	57	63
3	5603	94.2	59	58
4	6162	105	59	52
5	6648	109	61	49
6	7105	107	67	46

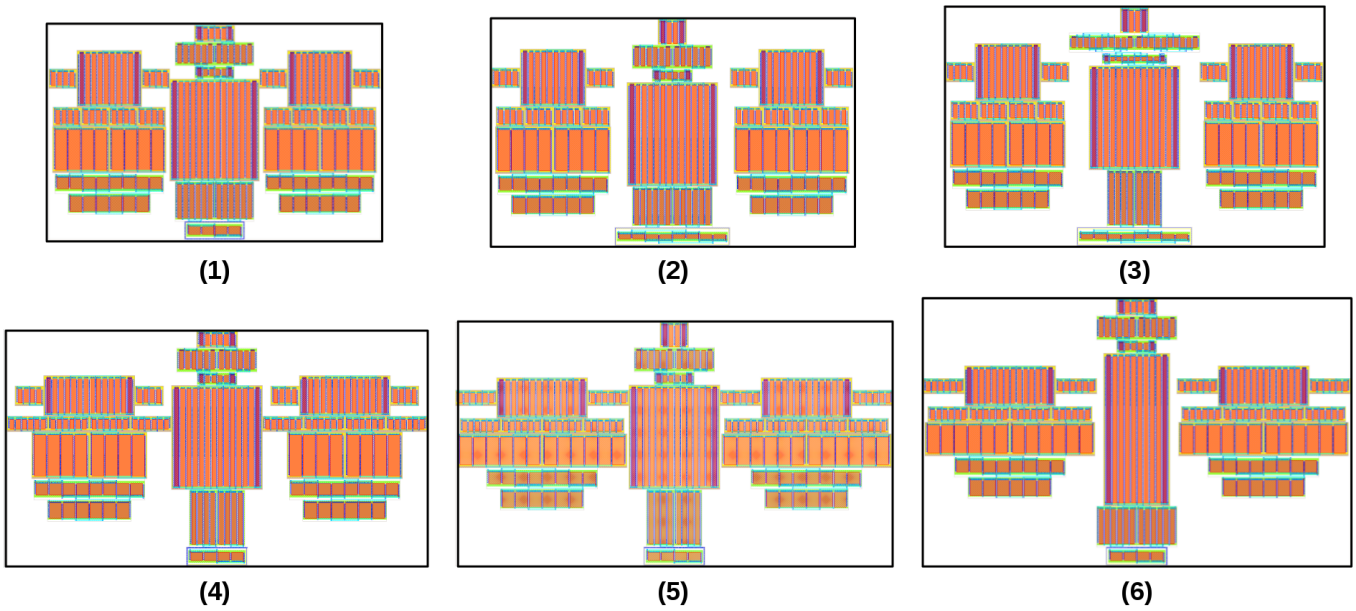

Fig. 5. Layouts of the fully differential transconductor [19] from Table I. These figures all have the same scale.

designed under a technology CMOS 130 nm.

The fully differential transconductor is composed of a total of 32 devices and we consider 2 possible variations for each device. The slicing tree takes into account 11 symmetries for this circuit and tolerance margins are set in a way to have reasonnable amount of accepted possibilities. Our algorithm found 384 possible placements in less than 1 second, some of them are listed on Table I and can be seen on Fig.5.

Table I shows the characteristics of those layouts with their total area, their width, their height and the pourcentage occupation of the circuit in the total area. Fig.4 illustrates a zoomed area of the first and the sixth placement from Table I in order to show the variation of fingers. Fig.5 illustrates the layouts described in Table I and show the evolution of the rows for different global aspect ratios. Designers can choose their final placement based on their experiences and preferences. Our tool presents the placement results plotted on a graph with heights and widths as axis and can be selected interactively to be placed in a few seconds.

V. Conclusion

We presented our semi-automatic analog placement approach using the slicing tree floorplan representation. The idea is to create an analog circuit organized in rows of devices where the acceptance of a row depends on a margin tolerance defined by the designer. By creating the slicing tree, the designer has more control over the placement phase than with an optimization-based approach. A placement solution is then selected according to height, width or global ratio criteria.

Slicing tree structure is a structure that has been studied over the last decades and its drawbacks are well-known. It is in a way restrictive since there are some topologies that cannot be represented with a slicing tree which can be a problem. Moreover, it does not extend to non-rectangular structures and has a bad white space distribution. Nevertheless, we are quite satisfied with the slicing tree structure which is able to handle the organization in row. For future considerations, even though the white space distribution can be a disadvantage, we plan to exploit it during the routing step that needs to be performed. Our floorplan representation and algorithm has been thought according to the analog routing step that we plan to develop in the future.

References

[1] M. P.-H. Lin, Y.-W. Chang and C.-M. Hung, "Recent Research and Development and New and Challenges and in Analog and Layout Synthesis," *in 21st Asia and South Pacific Design Automation Conference*, pp. 617-622, Jan. 2016.

[2] Helmut E. Graeb "Analog Layout Synthesis - A Survey of Topological Approaches," *in Springer Link*, 2011

[3] P.-H. Wu et al. "Exploring feasibilities of symmetry islands and monotonic current paths in slicing trees for analog placement," *in IEEE TCAD*, vol. 33, no. 6, pp. 879-892, Jun. 2014.

[4] P.-H. Lin, Y.-W. Chang and S.-C. Lin "Analog Placement Based on Symmetry-Island Formulation," *in Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, pp. 791-804, 2009.

[5] Q. Ma, E. Young, and K. Pun, "Analog placement with common centroid constraints," *in Proc. ICCAD*, pp. 579-585, 2007.

[6] M. Strasser et al., "Deterministic analog circuit placement using hierarchically bounded enumeration and enhanced shape functions," *in Proc. ICCAD*, pp. 306-313, 2008.

[7] D. Long, X. Hong and S. Dong "Signal-Path Driven Partition and Placement for Analog Circuit," *in Proc. ASP-DAC*, pp 694-699, 2006.

[8] M.-H. Lin, H. Zhang, M. Wong, and Y.-W. Chang, "Thermal-driven analog placement considering device matching," *in Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, pp. 325-336, 2011

[9] S. Nakatake, "Structured placement with topological regularity evaluation," *in Proc. ASP-DAC*, pp. 215-220, 2007.

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

[11] Q. Ma, L. Xiao, Y.-C Tam, E. F. Y. Young "Simultaneous Handling of Symmetry, Common Centroid, and General Placement Constraints," *in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30:1, pp.85-95, 2010

[12] N. Lourenço, A. Canelas, R. Póvoa, R. Martins and N. Horta, "Floorplan-aware analog IC sizing and optimization based on topological constraints," *in Integration, the VLSI Journal*, vol. 48, pp. 183-197, 2015

[13] R. He and L. Zhang, "Symmetry-aware TCG-based placement design under complex multi-group constraints for analog circuit layouts," *in Proc. ASP-DAC, 2010*, pp. 299-304, 2010

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

[15] A. Unutulmaz, G. Dündar and F. V. Fernández, "On the convex formulation of area for slicing floorplans," *in Integration, the VLSI Journal*, vol. 50, pp. 74-80, 2015

[16] *Project Cairo-Hurricane AMS (CHAMS)*, Laboratory LIP6, Pierre and Marie CURIE University, http://www-soc.lip6.fr/recherche/cian/chams/

[17] T.-C. Chen, T.-Y. Kuan, C.-C. Hsieh and C.-C. Peng, "Challenges and Solutions in Modern Analog Placement," *in VLSI Design, Automation, and Test (VLSI-DAT)*, pp. 1-4, April 2012.

[18] S. M. Saif, M. Dessouky, M. W. El-Kharashi, H. Abbas and S. Nassar, "Pareto front analog layout placement using Satisfiability Modulo Theories," *in DATE 2016*, pp. 1411-1416, 2016

[19] R. Iskander, M.-M. Louërat, A. Kaiser, "Hierarchical sizing and biasing of analog firm intellectual properties," *in Integration, the VLSI Journal*, pp. 172-188, 2013