

A Multi-level Analog IC Design Flow for Fast Performance Estimation Using Template-based Layout Generators and Structural Models

Benjamin Prautsch¹, Thomas Markwirth¹, Frank Schenkel², Reimund Wittmann³, Uwe Eichler¹, Jens Lienig⁴

¹Fraunhofer IIS/EAS, Institute for Integrated Circuits, Division Engineering of Adaptive Systems, Dresden, Germany
{Benjamin.Prautsch, Thomas.Markwirth, Uwe.Eichler}@eas.iis.fraunhofer.de

²MunEDA GmbH, München, Germany, Frank C. Schenkel frank.schenkel@muneda.com

³IMST GmbH, Kamp-Lintfort, Germany, reimund.wittmann@imst.de

⁴Dresden University of Technology, Dresden, Germany; jens@ieee.org

Abstract—Analog IC design is a very challenging task as essential information is missing in the early design stages. Because the simulation of larger designs is exceedingly computationally expensive at lower abstraction levels, conservative assumptions are usually applied which often result in suboptimal performances such as area and power consumption. In order to enable both early performance estimates and fast iteration cycles, we combined the estimation of parasitics from template-based layout generators with SystemC-based parameterizable modelling. As a result, we can compute layout-aware performance estimates of a configurable capacitive pipeline ADC within a runtime of only about one minute per iteration. Using this estimation in a loop, we analyzed and optimized substantial parameters of a capacitor array in order to improve the ADC’s performance.

Keywords—Analog Design, EDA, Layout, Generator, Template, Estimation, Parasitics, Layout-Aware, Optimization

I. INTRODUCTION

Due to the many objectives across a variety of abstraction levels, analog design is a very complex and multi-objective optimization problem making it very challenging and time-consuming. Therefore, many automation approaches tackle analog design using optimization methods. This way, e.g. electrical performances are traded-off against requirements in a layout-aware optimization flow [1].

Full Layout synthesis is possible without templates [2], however, many recent approaches employ them [3]. Templates are design methodological placement constraints that represent the layout in an abstract and machine-readable way. Therefore, they allow guidance of optimization that, in addition, can include fast routing estimates to yield better confidence in the simulated results [4].

Generators are required for layout creation, especially at lower design levels. They are built with procedural (implicit) code and create layout variants controlled by parameters [5]. One recent approach incorporates formalized knowledge and applies agent-based optimization [6]. Generators can also utilize templates for direct control of the layout automation in a machine-readable way [7, 8]. In [9] a strict layout grid is applied for systematic placement in advanced process nodes. Soft-IP-oriented generator approaches even create schematic, symbol or further views of a building block [10]. Some approaches trigger sizing tools in order to derive an optimized schematic-level performance [4, 11]. Specifically addressing capacitance arrays, [12] applies algorithms at the layout level in order to co-optimize common-centroid placement while reducing the routing parasitics.

We adopted the approach of early layout estimates and included it into our template-driven generator tool. Also, we implemented a parametric pipeline ADC model and incorporated both into a hierarchical tool chain for performance estimation of both the pure layout and an overall pipeline ADC design. To our knowledge, this is the first integrated combination of all aforementioned aspects into one automated tool chain that comprises:

- Template-based generator method,
- Parasitic estimation using object-oriented templates, and
- Fast model-based simulation.

Our approach enables early performance estimates by an integrated flow connecting generator, template, and model.

II. THE TOOL CHAIN

We combined a template-based generator approach and a SystemC AMS pipeline ADC model into an executable tool chain. An input parameter set configures both template (e.g. number of device rows) and model (e.g. non-ideal capacitors or OpAmp offsets). With the seamless integration of all these components into an executable flow, we can investigate the performance space of a multitude of parameter sets at low computational cost and find good tradeoff candidates fast. As shown in Figure 1, estimates of layout parasitics and the entire pipeline ADC are calculated fast in about five seconds and a minute, respectively. Subsequent generator execution with the most promising parameter set creates the related capacitor layout arrangement also in about a minute. Conventional manual layout design and full simulation would likely take a full working day plus several hours of simulation time.

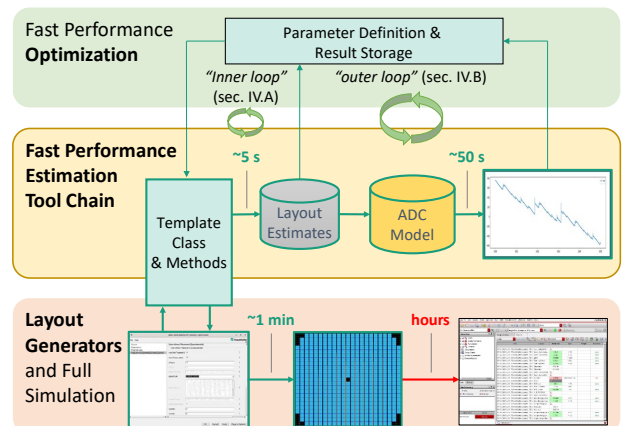


Figure 1 Tool chain with generator template, model, performance estimation (yellow) and its relation to both optimization flow (green) and full simulation flow (red). Major steps are framed green.

A. Parameter Extraction & Estimation

In order to realize a fast estimate in the template, the effective capacitance values of devices and routing wires were pre-characterized. We first extracted the capacitance values directly from the device pCells as a function of the sizing through a small script. These values are then included into a table model with first order interpolation. Additionally, we derived the sheet capacitance of the routing wires towards substrate from parasitic extraction of different configurations in width and spacing for the lowest metal to get conservative estimates. These parameters were then fitted to an analytical function that depends on both wire width and spacing to neighboring metals.

B. Generator Templates

Templates improve the flexibility of layout generators as they describe the layout in a more abstract way [8] and they centralize similar problem-specific code of (many) generators, like such for matrix arrangements [13], at a single place.

Our templates were implemented as part of our object-oriented generator programming interface and, thus, allow easy extension by further template classes for other specific arrangements. In addition to methods and properties for their original purpose—the relative positioning of layout elements—our templates contain methods for the estimation of parasitics that result from the specified arrangement, element types, and sizing. This allows to analyze individual abstract layouts within less than a second up to a few seconds.

Hence, the optimization does not require the actual layout generation step and solely the (fast) template is used during optimization. Once a promising parameter set was found, it can be passed directly to the generator to create the corresponding layout for detailed investigation through extraction.

C. Modelling

In order to reflect the behavior from device level on system level, we implemented a SystemC AMS model of a pipeline ADC. Its level of detail is selectable from behavioral to structural in order to trade-off runtime vs. accuracy. In the structural mode used in this work (the behavioral mode is used on the system level), individual elements are parameterizable, so that they mimic the parameters and performances of the actual circuits (especially capacitor values and, e.g., switch resistances and OpAmp offsets). This way, we obtain a similar behavior at faster simulation run time.

Based on initial device-level simulations of sub blocks of the ADC, the model parameters were determined and written into a JSON file. Because many of the model's subblocks directly take the wanted performance (e.g. gain, offset) as a parameter, no parameter fitting step is required for, e.g., the comparator capacitances of the individual pipeline stages in the system model, the switches, and the OpAmps. This has the advantage that parameter changes and a resulting modified parameter file do not require a new model translation and an equivalent system model is immediately available after characterization on transistor level.

While the analytical model from [14] allows thousand complete runs of a charge redistribution SAR ADC within only a few minutes, our detailed structural model incorporates

more detail, it is parameterizable, and it still runs reasonably fast in less than a minute.

D. Parameter Optimization Flow

We applied a hierarchical parameter optimization flow. This flow starts with a fast “inner” iteration loop and then progresses “outwards” to the iteration loop that incorporates the ADC model. Within the inner loop, an optimization algorithm is combined with the stand-alone template in order to determine reasonable trade-offs at the pure layout level (see section IV.A). The outer loop incorporates both template and ADC model in order to calculate the ADC's electrical performance given the influence of the capacitor array.

The optimized input parameters of the template can finally be fed into the generator as is. As these parameters control the generation process of the layout, it can then be generated, extracted, and simulated using the designer's gold-standard simulator.

III. EVALUATION METHODS

The generator template implements methods to estimate layout parasitics, capacitor variation, and mismatch. With the generator parameters as input, it enables fast analyses. Once computed, an output file is generated storing information such as effective capacitances estimates for further use in the flow.

A. Parasitic Layout Effects & Layout Optimization

The effective capacitor ratio is crucial for sensitive analog designs. Therefore, we estimate the parasitic capacitance of the routing wires C_{rt} in the array and assign it to the logical capacitor to which the wires connect. The sum of the layout's unit capacitors C_{dev} belonging to a logical capacitor are also assigned. This way, the template collects all $M(n)$ sub capacitance values from both unit devices and routing estimates of each logical device n out of N logical devices:

$$C_{rt\ n} = \sum_{i=1}^{M(n)} C_{rt\ sub\ i}, \quad n = 1, \dots, N, \quad (1)$$

$$C_{dev\ n} = \sum_{i=1}^{M(n)} C_{dev\ sub\ i}, \quad n = 1, \dots, N. \quad (2)$$

Based on the capacitance values, we derive two figures of merit (FOM) that (1) quantize the ratio of the parasitic routing capacitance to the logical device capacitance (ideally equal to zero) and (2) measure the error of the effective capacitance including routing to the targeted ideal capacitance ratio of the ideal capacitor (ideally equal to one):

$$FOM_1 = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(0 - \frac{C_{rt\ i}}{C_{dev\ i}}\right)^2}, \quad (3)$$

$$FOM_2 = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(1 - \frac{C_{dev\ i} + C_{rt\ i}}{C_{ideal\ i}}\right)^2}. \quad (4)$$

B. Mismatch and Corners

Every integrated device differs from its ideal nominal behavior through both *global* variation and *local* mismatch effects [15, 16] which necessitates high precision design techniques for converters [17]. In their largely adopted work, M. Pelgrom et al. showed [15] that the variance of a parameter ΔP between two rectangular transistor devices derives to:

$$\sigma^2(\Delta P) = \frac{A_p^2}{WL} + S_p^2 D_x^2. \quad (5)$$

The effect of mismatch through the parameter A_p is, therefore, controllable by both width W and length L of a device while the parameter S_p can be diminished through small distances D_x between devices. Both A_p and S_p depend on the particular technology and device used. Practically, S_p is not always provided by the PDK. Moreover, an effective value of D_x must be derived in the often occurring case that more than two devices are to be matched with each other. Usual methods are computationally expensive with $O(N^2)$ which motivates a physically motivated model for efficient CAD [18]. We did also consider comparing each unit device with an assumed ideal device at the common centroid point which would result in only $O(N)$ calculations. In our actual analysis, however, due to the assumed small distances in our data converter applications, we can likely neglect the effect of S_p [17].

Assuming independent random influences, the effects of mismatch and corners is distributed according to the Gaussian normal distribution:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \quad (6)$$

From the PDK, we usually get the parameter information of minimal (at -3σ), typical (at $x = \mu$), and maximum variation (at 3σ) as well as parameter A_p while parameter S_p , however, might not be available.

C. Overall Error Quantification

In order to take both global and local process variation into account, we combined minimal, typical, and maximum capacitance value with the local mismatch as well as with the influence from parasitic routing. Beforehand, the effective capacitance difference from mismatch is derived from the variance of the capacitor:

$$\sigma\left(\frac{\Delta C}{C}\right) = \frac{A_p}{\sqrt{WL}} \xrightarrow{C=const.} \sigma(\Delta C) = \frac{A_p * C}{\sqrt{WL}}. \quad (7)$$

With equation (7), we can combine either of the global 3σ corner case values $C_{dev\ cor}$ with its related local device mismatch from the layout in order to form worst cases of the effective device capacitance value:

$$C_{dev\ wc} = C_{dev\ cor} \pm 3 \frac{A_p * C_{dev}}{\sqrt{WL}}. \quad (8)$$

By including minimum and maximum worst cases into equations (3) and (4), respectively, the worst-case device capacitance can be considered together with the routing capacitances C_{rt} during the evaluation.

IV. EXPERIMENTAL RESULTS

In order to quantify both performance estimates and runtime, we applied the proposed tool chain to the pipeline ADC model mentioned in section II.C. The actual ADC was first designed at transistor level. Subsequently, we built a parameterizable model on this basis. This way, fast simulation and optimization with estimated values becomes possible. The major targets of the optimization are (1) reduced area, (2)

fitting aspect ratio, and (3) minimal error in the effective capacitor ratio with good robustness against process variations and mismatch.

A. Optimization at Layout Level

The goal of optimization strategies [19, 20] is to find values for design parameters \mathbf{d} (i.e. adjustable parameters that influence the behavior $f(\mathbf{d})$ of a circuit or system) to meet given specifications on the circuit performances f :

$$f_L \leq f(\mathbf{d}) \leq f_U \quad (9)$$

with the vector of performance measures $\mathbf{f} = (f_1, \dots, f_n)$ and the vectors of lower bounds \mathbf{f}_L and upper bounds \mathbf{f}_U . When optimizing the layout of our capacitor array, the design parameters \mathbf{d} and performances \mathbf{f} are considered as follows:

- Design parameters are the height and the width of a capacitor unit device and the number of rows in the array of unit devices. All parameters for both switches and OpAmps are fixed.
- Performances evaluated comprise the overall area of the capacitor array, a measure of the effective error introduced by the routing parasitics FOM_1 , an error measure of the accuracy of the targeted capacitance ratio FOM_2 , and the array's aspect ratio.

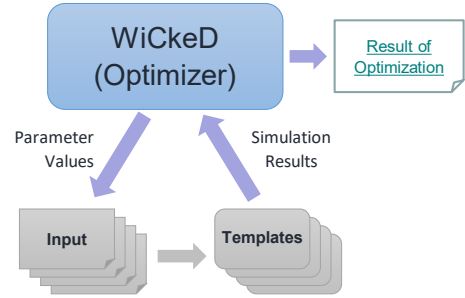


Figure 2 Optimization flow using WiCkeD. After initial configuration for optimization, WiCkeD iteratively generates input parameters, runs the template, evaluates the results, and restarts in order to find parameter values for optimized results.

The layout-level optimization was done with WiCkeD from MunEDA [21]. WiCkeD uses a classical *simulator-in-a-loop* approach with the template in this case serving as a simulator (Figure 2). Whenever the optimizer needs to determine the performance(s) \mathbf{f} for a given set of design parameters \mathbf{d} , it calls the template to evaluate them. This holds true even for gradient calculations, needed for the optimizer to determine the direction in the space of design parameters in which to improve the layout.

Due to short run time to evaluate the template (≤ 5 s), the optimization algorithm quickly returned an optimized set of layout parameters. In order to implicitly consider matching (distance term in Pelgrom's law), we added a penalty on layout aspect ratios being far from 1:1.

B. Evaluation of Pipeline ADC Performance Measures

In order to evaluate the estimated ADC performance, we utilized the reduced input parameter set identified in section IV.A and evaluated the model outputs through the tool chain. The input parameter range of W and L of the unit devices and the number of rows $nRows$ in the array were

selected such that the shape of the resulting capacitor array leads to a reasonable aspect ratio. With these parameters, a simulation study of the worst case performances incl. offsets was run in less than two hours with 114 individual parameterizations. A representative subset of them is given in Figure 3 and exemplary worst-case error measures (offset and DNL) from all runs are shown in Figure 4. The error largely depends on the area ($W * L$) as expected from equations (3) and (4), as C_{dev} and C_{ideal} are proportional to it. The number of rows and the changing ratio of W and L adds additional variance to the performance estimates.

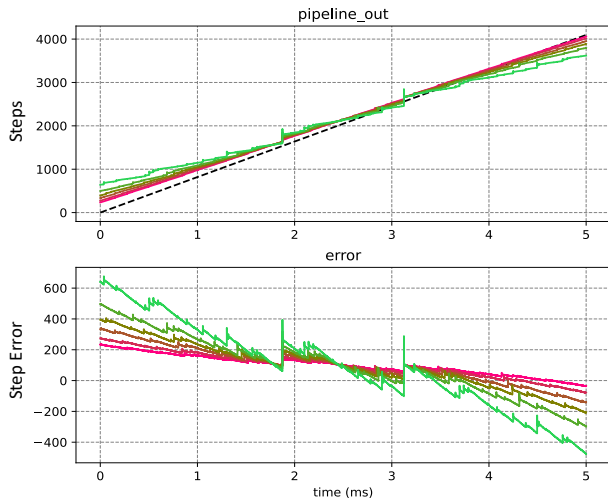


Figure 3 Representative subset of the worst-case transfer functions of the ADC model (with 4096 steps being full-scale) that were analyzed for different numbers of rows in the array and different ratios of W over L while varying the area $W * L$. The area has a major effect represented by the selected parameter subset in this figure. The ideal curve is dashed.

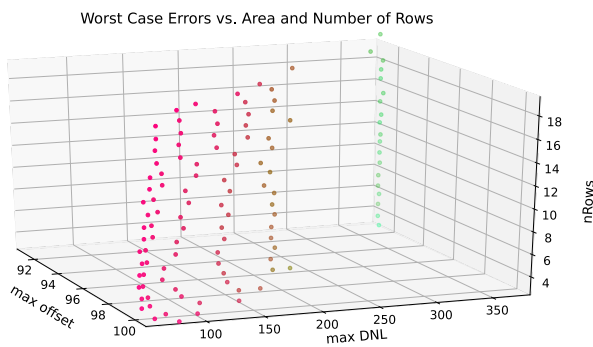


Figure 4 Worst-case error estimates of exemplary ADC performances for all parameter sets. The number of rows ($nRows$) shows only a minor impact on the error figures caused by different placement and related routing. Depending on the device area, maximum offset and the maximum differential nonlinearity (DNL) change significantly and relevant trade-offs can be selected.

V. CONCLUSION AND OUTLOOK

In this work, we proposed a new tool chain for fast performance estimation that integrates generators, template-based layout estimates, and a structural model of a pipeline ADC. We used our tool chain for parameter optimization of the capacitor layout in a pipeline ADC. First, a fast inner loop was used to optimize the layout-level estimates derived directly from the template. In a second, larger loop, the template's estimates were fed to a previously built parameterizable model in order to estimate the ADC's

performance. With the inner loop running about five seconds and the outer loop running just below a minute, one can identify the fitting layout design parameters fast.

We believe that this approach is a valuable tool for circuit designers in order to create estimates relatively quickly, thereby identifying promising parameters for direct use in the subsequent generator-based design flow.

Future work could incorporate more layout analyses like on crosstalk, matching (e.g. S_p), or further layout effects like line edge roughness of wires. Also, statistical optimization in order to increase the overall yield could be considered.

ACKNOWLEDGMENT

This work was enabled by the project AnastASICA (grants 16ES0990, 16ES0988K, 16ES0989), funded by the German Federal Ministry of Education and Research (BMBF).

REFERENCES

- [1] N. Lourenço et al., "AIDA: Robust Analog Circuit-Level Sizing and In-Loop Layout Generation," *Integration, the VLSI Journal*, 2016.
- [2] H. Habal and H. Graeb, "Constraint-based Layout-driven Sizing of Analog Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30.8, pp. 1089-1102, 2011.
- [3] H. E. Graeb, *Analog Layout Synthesis. A Survey of Topological Approaches*, Springer, 2011. DOI: 10.1007/978-1-4419-6932-3.
- [4] N. Lourenço, R. Martins and N. Horta, "Layout-aware Sizing of Analog ICs Using Floorplan & Routing Estimates for Parasitic Extraction," *Design, Automation & Test in Europe Conf. & Exhibition (DATE)*, pp. 1156-1161, 2015.
- [5] J. Scheible and J. Lienig, "Automation of Analog IC Layout – Challenges and Solutions," *Proc. of the ACM 2015 Int. Symp. on Physical Design (ISPD'15)*, pp. 33-40, March 2015.
- [6] D. Marolt, *Layout Automation in Analog IC Design with Formalized and Nonformalized Expert Knowledge*, Dissertation, Stuttgart, 2018.
- [7] B. Prautsch, U. Hatnik, U. Eichler und J. Lienig, "Template-Driven Analog Layout Generators for Improved Technology Independence," *Proc. of ANALOG 2018*, pp. 156-161, Sept. 2018. ISBN: 978-3-8007-4754-2.
- [8] B. Prautsch, R. Wittmann, U. Eichler, U. Hatnik and J. Lienig, "Generators, Templates, and Code Generation for Flexible Automation of Array-Style Layouts," *Proc. of the Int. Conf. on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD 2021)*, pp. 180-183, July 2021.
- [9] J. Han et al., "LAYGO: A Template-and-Grid-based Layout Generation Engine for Advanced CMOS Technologies," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 3, pp. 1012-1022, Mar. 2021, doi: 10.1109/TCAD.2020.3046524.
- [10] B. Prautsch et al., "IIP Framework: A Tool for Reuse-centric Analog Circuit Design," *13th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pp. 1-4, June 2016.
- [11] 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," *IEEE Custom Integrated Circuits Conf. (CICC)*, pp. 1-8, 2018, doi: 10.1109/CICC.2018.8357061.
- [12] M. P. Lin, Y. He, V. W. Hsiao, R. Chang and S. Lee, "Common-Centroid Capacitor Layout Generation Considering Device Matching and Parasitic Minimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 7, pp. 991-1002, Jul. 2013, doi: 10.1109/TCAD.2012.2226457.
- [13] B. Prautsch, U. Eichler, T. Reich und J. Lienig, "MESH: Explicit and Flexible Generation of Analog Arrays," *2017 14th Int. Conf. on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pp. 1-4, July 2017. DOI: 10.1109/SMACD.2017.7981572.
- [14] S. Haenzsche, S. Henker and R. Schüffny, "Modelling of Capacitor Mismatch and Non-Linearity Effects in Charge Redistribution SAR ADCs," *Proc. of the 17th Int. Conf. Mixed Design of Integrated Circuits and Systems - MIXDES*, pp. 300-305, 2010.
- [15] M. J. Pelgrom, A. C. Duinmaijer and A. P. Welbers, "Matching Properties of MOS Transistors," *IEEE Journal of solid-state circuits*, vol. 24, no. 5, pp. 1433-1439, 1989.
- [16] S. Dongaonkar, S. P. Mudanai and M. D. Giles, "From Process Corners to Statistical Circuit Design Methodology: Opportunities and Challenges," *IEEE Trans. on Electron Devices*, vol. 66, no. 1, Jan. 2019.
- [17] R. Wittmann, W. Schardein, B. J. Hosticka, G. Burbach and J. Arndt, "Trimless High Precision Ratioed Resistors in D/A and A/D Converters," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, Aug. 1995, doi: 10.1109/4.400436.
- [18] B. Linares-Barranco and T. Serrano-Gotarredona, "A Physical Interpretation of the Distance Term in Pelgrom's Mismatch Model Results in Very Efficient CAD," *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pp. 1561-1564, 2007, doi: 10.1109/ISCAS.2007.378710.
- [19] G. Quarata and M. Pronath, "Analysis and Optimization of Voltage Reference Circuits Based on Sub-1V MOSFETs Operating in Different CMOS Technologies," *16. GMMITG-Fachtagung Analog ANALOG, VDE*, pp. 1-6, Sept. 2018.
- [20] F. Cacho, A. Cros, X. Federspiel, V. Huard and C. Roma, "BTI Induced Dispersion: Challenges and Opportunities for SRAM Bit Cell Optimization," *IEEE Int. Reliability Physics Symp. (IRPS)*, 2016.
- [21] MunEDA, "WiCkeD EDA Tools for Process Migration, Sizing, and Verification of Custom IC," [Online]. Available: <https://www.muneda.com>. [Accessed 28 04 2022].