HAL
open science

A Seamless Representation for Coupling Transistor Sizing with Nanometric CMOS Layout Generation

Stéphanie Youssef, Farakh Javid, Damien Dupuis, Ramy Iskander, Marie-Minerve Louërat

▶ To cite this version:

Stéphanie Youssef, Farakh Javid, Damien Dupuis, Ramy Iskander, Marie-Minerve Louërat. A Seamless Representation for Coupling Transistor Sizing with Nanometric CMOS Layout Generation. 20th European Conference on Circuit Theory and Design (ECCTD), Aug 2011, Linkoping, Sweden. pp.341-344, 10.1109/ECCTD.2011.6043356 . hal-00749892

HAL Id: hal-00749892

https://hal.science/hal-00749892v1

Submitted on 8 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Seamless Representation for Coupling Transistor Sizing with Nanometric CMOS Layout Generation

Stéphanie Youssef, Farakh Javid, Damien Dupuis, Ramy Iskander and Marie-Minerve Louerat
University Pierre et Marie Curie (UPMC)
Computer Science Laboratory (LIP6)
4, Place Jussieu, 75005 Paris, France
Email : stephanie.youssef@lip6.fr

***Abstract*—In this paper, a new method for developing smart parameterized generators for analogue devices is presented. A device is an atomic analogue cell that performs an elementary and standard function such as the differential pair and the current mirror. A device is smart since it can be electrically and physically adapted. In the proposed method, the device sizes and biases are first computed using dedicated *sizing operators* based on the MOS transistor model and the foundry Design Kit. Once transistor sizes are computed, they are fed to a layout generation tool which offers different layout styles for the same device. The layout is generated with the layout dependent parasitics, including stress effects. These parasitics are then taken into account by the sizing operators. Therefore a loop between sizing and layout generation can be set and executed until the device specifications are met. The method is applied to a differential pair with several layout styles and two distinct technologies.**

I. Introduction

An ultimate objective for today's analog EDA flows is to provide a library of atomic analogue cells integrating a tight link between accurate electrical sizing and nanometric layout generation. These atomic cells should provide a wide range of layout styles and aspect ratios. It is expected that these libraries can handle efficiently the problem of technology porting. Solving the problem of porting is of great interest to the mixed-signal System-On-Chip market. Lots of studies have proposed partial solutions to the porting problem. Some studies focused on the layout-aware issues [1]–[7], others on technology porting [8]–[13], others on the MOS modeling [14]–[17] and their relations to the stress and proximity effects [18]–[20].

To our knowledge, very few number of tools provide the designer with a fast and accurate way to realize different layouts for the same analogue atomic function. A remarkable progress has been made by CIRANOVA [21] which develops parameterized cells in Python Language, known as *PyCells*. PyCells deliver physical views in OpenAccess [22] which is Cadence standard and interoperable database. Many EDA startups have invested in developing OpenAccess native applications. OpenAccess is currently being pushed as a standard database for the EDA industry. OpenAccess assures interoperability, speed, usability. Yet, interoperability standards still need to be agreed upon for this to be true.

In this paper, we propose a novel methodology that allows to size and bias an atomic analogue cell (device) and generate different layout styles very seamlessly. The idea is to well characterize the electrical and the physical parameters of the device to meet functional and robustness constraints [23]. Based on [24], sizing and biasing operators are coupled in a loop with very fast nanometric layout generation tool that allows to describe device layouts in Python. The sizing operators propose sizes to the layout generation tool. This in turn realizes the layout for a given style. Then it computes physical sizes, stress effects as well as layout dependent parameters. These are then fed back to the sizing operator to be taken into account in the next iteration. We show that the flow is very simple and achieves satisfactory results in a matter of seconds. The advantage of the proposed flow is to couple seamlessly and in a procedural manner both transistor sizing and nanometric layout generation, with strong focus on the device intrinsic performance.

The paper is organized as follows: section II introduces the device definition and the coupling between sizing and layout generation. Section III presents the results of the technology migration from 130 nm CMOS technology to 65nm CMOS technology for different layout styles. Finally, section IV concludes the paper.

II. Coupling Sizing and Layout

A. The device sizing and layout features

1) Device Definition: A device is defined as an atomic analogue function realized by a small set of transistors. The motivation to build a device is the following: the analogue electrical behavior of the set of transistors requires a dedicated layout with strong geometrical and robustness constraints. Therefore the layout of the transistor's set has to be designed as a whole. A typical library of analogue devices contains: a folded transistor, a differential pair, a current mirror and a cross coupled pair. Each device may have different layout styles. Here we will study four different styles of the differential pair in Fig.1: interdigitated, symmetrical, 2D common centroid and M2 module [1].

The goal of the device is to provide an electrical realization along with a physical realization (layout) of an atomic behavior annotated with all the layout dependent parameters.

The device is a smart object, since it has two main features: on one hand a set of methods to study the electrical behavior and on the other hand a set of methods to generate a layout.

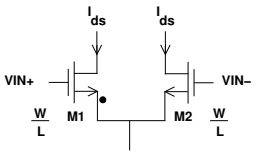

Fig. 1. Differential Pair

2) Electrical Operation: To offer a large possible choice to the designer, the device comes with an electrical API. This API consists of a set of operators. These operators have two goals: the first set is dedicated to size the device according to some input specification and the second one is dedicated to analyze in details the electrical behavior of the device, taking into account all the details of physical realization.

Sizing and biasing operators are based on the transistor compact model equations. Operators are used in both *sizing* and *analysis* phases. In the sizing phase, the operator computes unknown widths and biases (Table I, where $V_{EG} = V_{GS} - V_{TH}$) according to input parameters set by the designer. A sizing operator computes either $W = f_W(Temp, I_{DS}, L, V_{GS}, V_{DS}, V_{BS})$, or $V_{GS} = f_{V_{GS}}(Temp, W, L, I_{DS}, V_{DS}, V_{BS})$, where f_W and $f_{V_{GS}}$ are two partial inverse functions of the compact model $I_{DS} = f_{MODEL}(Temp, W, L, V_{GS}, V_{DS}, V_{BS})$. *MODEL* is a standard transistor model like BSIM3V3, BSIM4, PSP, EKV. Sizing operators use simulator encapsulation [24], ensuring accurate computed results. During the analysis phase, the OPIDS operator is used to compute the current as well as the small signal parameters taking into account the layout dependent parameters.

TABLE I
CLASS DEFINITION OF SIZING & BIASING OPERATORS

Operator	Definition
$OPVS(V_{EG}, V_B)$	$(Temp, I_{DS}, L, V_{EG}, V_D, V_G, V_B) \mapsto (V_S, W, V_{TH})$
...	...
$OPVG(V_{EG})$	$(Temp, I_{DS}, L, V_{EG}, V_D, V_S) \mapsto (V_G, W, V_{TH}, V_B)$
...	...
$OPVGD(V_{EG})$	$(Temp, I_{DS}, L, V_{EG}, V_S) \mapsto (V_G, V_D, W, V_{TH}, V_B)$
...	...
$OPW(V_G, V_S)$	$(Temp, I_{DS}, L, V_D, V_G, V_S) \mapsto (W, V_{TH}, V_B)$
...	...
$OPIDS(V_G, V_S)$	$(Temp, W, L, V_D, V_G, V_S) \mapsto (I_{DS}, V_{TH}, V_B)$
...	...

3) Layout generation: The sizing phase in the device design process results in electrical sizes width and lengths (W_e and L_e). These data are then used to generate the layout. The shape of the device layout is controlled by the following parameters: W_{ph}, L_{ph}, layout style (interdigitated, symmetric, 2D common-centroid, M2 module) and the process design rules. A dedicated Python API has been developed to describe the device layout. For each device, in addition to the method describing the layout, three special methods have been developed to compute the layout dependent parameters of the MOS transistor model. The first one computes the area and perimeter of the source and drain zones, the second one computes the stress effect parameters introduced in the BSIM4 model and the third one computes the capacitances of the routing wires. The layout Python API offers the possibility to describe technology independent layout generators.

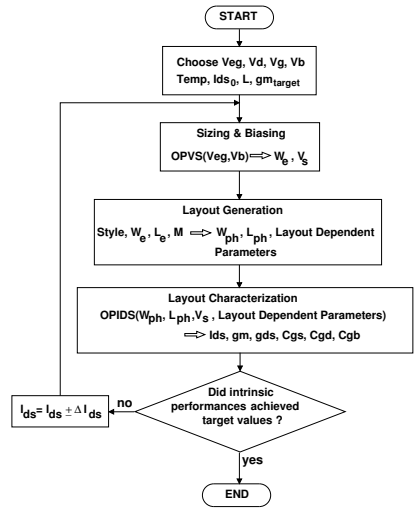

Fig. 2. Sizing and layout generation design flow

B. The design flow

Figure 2 illustrates how the operators and the layout method are used to implement a device while respecting specifications. In the first step, given the temperature, the biasing current I_{DS}, the overdrive gate voltage V_{EG}, the drain voltage V_D, the gate voltage V_G, the bulk voltage V_B and the transistor length L, the operator OPVS is used to compute the electrical width W_e, the source voltage V_S and the threshold voltage V_{TH}. The width W_e, length L_e, number of fingers M and layout style are given to the layout generator. Once the layout is generated, the actual physical width W_{ph} and length L_{ph} as well as the layout dependent parameters (diffusion zone, stress and routing) are available. An accurate characterization, including the actual physical realization, is performed using the OPIDS operator that provides the actual I_{DS} and the small signal parameters for the purpose verification.

Here, we choose to take the g_m as a specification in the case of the differential pair. If the g_m value does not meet the specification, a loop is set to adjust the biasing current till the specifications are achieved. After convergence, the final layout is then realized.

Note that in the flow, all parameters varies, except the layout style and the number of fingers that are kept invariant in this loop.

III. RESULTS

A. Differential Pair Design and Migration

In this section, we present some design results of the differential pair (Fig. 1) using the flow of Fig. 2 in CMOS 65nm technology.

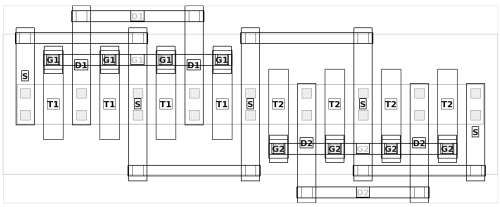

Fig. 3. Differential Pair (interdigitated layout style) 65nm

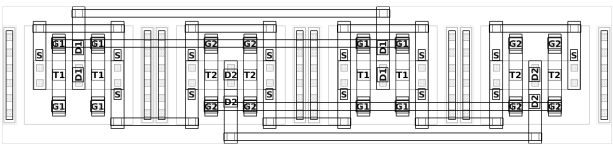

Fig. 4. Differential Pair (M2 module layout style) 65nm

First, we start to design a differential pair in a CMOS 130nm technology. The first goal is to design a differential pair with the assumption: I_{DS}, V_{EG},V_D, V_G, V_B and the transistor length L (set to $3.L_{min}$), are known. The three steps of the flow (Fig. 2) are executed without the need to use the loop. The design is performed for 4 different layout styles with 4 fingers. The resulting g_m is computed, including the effect of the physical realization and equal to $0.37m.S$.

Then, the technology porting is performed. Using the same design flow (Fig. 2), with the same operators and the same layout generator and changing the technology files by the CMOS 65nm ones (transistor MOS model and design rules), the goal is to design a differential pair with the same target g_m obtained for the CMOS 130nm technology. Some authors [8]–[13] have tried to solve the technology porting issues by guessing some scaling parameters and deriving the new transistors sizes accordingly. Such scaling is not appropriate in the case of nanometric MOS technologies. Rules to define scaling parameters are not easy to elaborate and we experienced that the design flow (Fig. 2), with loop execution, is a more accurate solution. The results of the loop execution is presented in the case of the four layout styles, Table II for the interdigitated style, Table III for the symmetrical, Table IV for the M2-module style and Table V for the 2D common-centroid. The $Iter$ parameter is the number of iteration in the loop, W_{ph} is the transistor width, I_{DS} is the drain current biasing the transistor, g_m is the transistor transconductance and sa and sb are the stress parameters defined by the BSIM4 model. The corresponding layouts are presented in Fig. 3 (interdigitated), Fig. 4 (M2 module), Fig. 5 (2D common-centroid) and Fig. 6 (symmetrical) respectively.

The tables show that the differences between the layout styles, resulting in different layout dependent parameters, have been well captured by the flow. Yet, for a given layout style, a given number of transistor's fingers, a given transistor's length, the lateral stress parameters are constant versus the finger width.

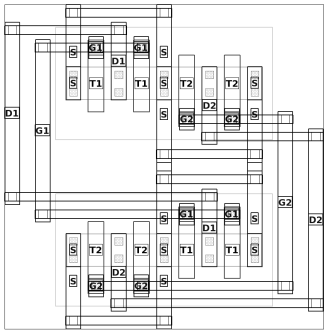

Fig. 5. Differential Pair (2D common centroid layout style) 65nm

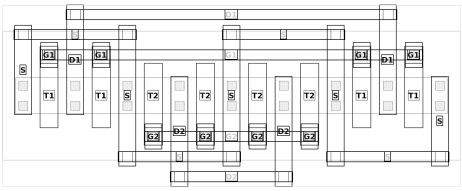

Fig. 6. Differential Pair (symmetrical layout style) 65nm

TABLE II
RESULTS 65NM INTERDIGITATED

Iter	$W_{ph}(\mu m)$	$I_{DS}(\mu A)$	$g_m(m\Omega^{-1})$	$sa(\mu m)$	$sb(\mu m)$
1	0.9	29.37	0.245	2.22	0.81
2	0.93	29.98	0.25	2.22	0.81
27	1.43	43.64	0.364	2.22	0.81
28	1.45	44.19	0.369	2.22	0.81

TABLE III
RESULTS 65NM SYMMETRICAL

Iter	$W_{ph}(\mu m)$	$I_{DS}(\mu A)$	$g_m(mS)$	$sa(\mu m)$	$sb(\mu m)$
1	0.9	29.60	0.247	1.99	1.99
2	0.93	30.30	0.253	1.99	1.99
26	1.41	43.62	0.363	1.99	1.99
27	1.43	44.18	0.368	1.99	1.99

TABLE IV
RESULTS 65NM M2

Iter	$W_{ph}(\mu m)$	$I_{DS}(\mu A)$	$g_m(mS)$	$sa(\mu m)$	$sb(\mu m)$
1	0.9	28.24	0.236	0.48	0.48
2	0.94	29.07	0.243	0.48	0.48
29	1.485	43.18	0.362	0.48	0.48
30	1.505	43.74	0.366	0.48	0.48

B. Differential Pair Migration Analysis

Let us take the example of the differential pair with 2D Common Centroid style Table V. After all the loop iterations to get the required g_m, we can get many information about the electrical and the physical parameters in CMOS 65 nm technology such as the electrical intrinsic capacitances: $C_{gs} = 2.39fF$, $C_{ds} = 1.16fF$, $C_{gd} = 0.558fF$, $C_{gb} = 0.37fF$, etc

TABLE V
RESULTS 65NM 2D-COMMONCENTROID

Iter	$W_{ph}(\mu m)$	$I_{DS}(\mu A)$	$g_m(mS)$	$sa(\mu m)$	$sb(\mu m)$
1	0.9	28.92	0.241	1.59	0.48
2	0.93	29.52	0.247	1.59	0.48
28	1.45	43.43	0.363	1.59	0.48
29	1.47	43.96	0.368	1.59	0.48

TABLE VI
RESULTS 130NM VS 65NM (2D COMMON CENTROID)

	130nm	65nm
$W_{ph}(\mu m)$	4.625	1.47
$L_{ph}(\mu m)$	0.39	0.18
$I_{DS}(\mu A)$	28.82	43.96
$g_m(mS)$	0.369	0.367
$C_{gs}(fF)$	13.63	2.39
$C_{gd}(fF)$	2.3	0.55
$C_{gb}(fF)$	1.17	0.37

... Also the physical parasitic routing capacitances: $C_{g_{ph}} = 0.0581fF$,$C_{d_{ph}} = 0.0937fF$,$C_{s_{ph}} = 0.03.89fF$, etc ... related to the net gate 'G', drain 'D' and source 'S' respectively. We can calculate similarly the transition frequency defined as:

$$F_t = \frac{g_m}{2\Pi(C_{gs} + C_{gd} + C_{gb} + C_{g_{ph}} + C_{s_{ph}})} = 17.6GHz$$

The overall computation time for the optimization loop is around 6s (1s for the sizing, 5s for layout generation). The average number of iterations is less than 30.

Sizing and migration results are compared in Table VI for 2D common centroid layout style. The computed width is decreased while migrating from 130 nm to 65 nm, however the current increases. Capacitances are also decreased. g_m is maintained during migrating.

IV. CONCLUSION

In this paper, a new method for developing smart parameterized generators for analogue devices has been presented. The interaction between the transistor sizing and the layout generation of the device has been illustrated in the case of the differential pair. Four different layout styles have been compared. The tight coupling between the transistor sizing and the layout generation has been used to solve the process migration challenge for devices. The proposed method allows the designer to select the most convenient device layout style for given specifications. The results showed the efficiency of the proposed method. As a future work, the proposed method will be extended to allow a seamless coupling between circuit level sizing and layout generation, taking into account intrinsic device performance.

REFERENCES

[1] L. Lewyn and N. Williams, "Is a new paradigm for nanoscale analog cmos design needed?" *Proceedings of the IEEE*, vol. 99, no. 1, pp. 3 –6, 2011.

[2] E. Yimaz and G. Dundar, "Analog layout generator for cmos circuits," *IEEE Trans. Computer-Aided Design*, vol. 28, no. 1, pp. 32–45, Jan. 2009.

[3] R. Castro-Lopez, O. Guerra, E. Roca, and F. Fernandez, "An integrated layout-synthesis approach for analog ics," *IEEE Trans. Computer-Aided Design*, vol. 27, no. 7, pp. 1179–1189, Jul. 2008.

[4] M. Ranjan, W. Verhaegen, A. Agarwal, H. Sampath, R. Vemuri, and G. Gielen, "Fast, Layout-Inclusive Analog Circuit Synthesis using Pre-Compiled Parasitic-Aware Symbolic Performance Models," in *Proc. Design Automation and Test in Europe Conf. (DATE)*, Feb. 2004, pp. 604–609.

[5] Tang, Zhang, and Doboli, "Layout-Aware Analog System Synthesis Based on Symbolic Layout Description and Combined Block Parameter Exploration, Placement and Global Routing," in *IEEE Symposium on VLSI, (ISVLSI)*, 2003.

[6] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic, and D. Leenaerts, "AMGIE– A Synthesis Environment for CMOS Analog Integrated Circuits," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 9, pp. 1037–1058, Sep. 2001.

[7] R. Naiknaware and T. Fiez, "Automated hierarchical CMOS analog circuit stack generation with intramodule connectivity and matching considerations," *IEEE J. of Solid-State Circuits*, vol. 34, no. 3, pp. 304–317, Mar. 1999.

[8] Alessandro Savio and Luigi Colalongo and Michele Quarantelli and Zsolt M. Kovacs-Vajna, "Automating Scaling Procedures for Analog Design Reuse," *IEEE Trans. on Circuits and Systems*, vol. 53, no. 12, pp. 2539–2547, Dec. 2006.

[9] T. Levi, N. Lewis, J.Tomas, and P. Fouillat, "Scaling Rules For Mos Analog Design Reuse," in *Proc. Int. Conf. Mixed Design of Integrated Circuits and Systems (MIXDES)*, vol. 1, Jun. 2006, pp. 378–382.

[10] S. Hammouda, H. Said, M. Dessouky, M. Tawfik, Q. Nguyen, W. M. Badawy, H. M. Abbas, and H. I. Shahein, "Chameleon art: a non-optimization based analog design migration framework," in *Proc. Design Automation Conf.*, 2006, pp. 885–888.

[11] Bhattacharya, Jangkrajarng, Hartono, and Shi, "Correct-by-Construction Layout-Centric Retargeting of Large Analog Designs ," in *Proc. Design Automation Conf.*, Jun. 2004, pp. 139–144.

[12] C. Galup-Montoro, M. Cherem Schneider, and R. Matos Coitinho, "Resizing rules for mos analog-design reuse," *IEEE Des. Test*, vol. 19, no. 2, pp. 50–58, 2002.

[13] K. Francken and G. Gielen, "Methodology for Analog Technology Porting Including Performance Tuning," in *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, vol. 1, May 1999, pp. 415–418.

[14] Ana Isabela Araujo Cunha and Marcio Cherme Schneider and Carlos Galup-Montoro, "A Transistor Model for Analog Circuit Design," vol. 33, no. 10, Oct. 1998, pp. 1510–1519.

[15] D. Binkley, *Tradeoffs and Optimization in Analog CMOS Design*. John Wiley and Sons, Inc., Jun. 2008, ISBN 978-0-470-03136-0.

[16] W. Liu, *MOSFET Models for SPICE Simulation including BSIM3v3 and BSIM4*. Wiley Interscience, 2001.

[17] C. C. Enz and E. A. Vittoz, *Charge-Based MOS Transistor Modeling: The EKV Model for Low-Power and RF IC Design*. John Wiley and Sons, Inc., Jul. 2006, ISBN 978-0-470-85541-6.

[18] J. X. *et al*, "Layout-dependent sti stress analysis and stress-aware rf/analog circuit design optimization," in *iccad*, San Jose, USA, November 2009, pp. 521–528.

[19] L. Y. *et al*, "Simulation of layout-dependent STI stress and its impact on circuit performance," in *Int. Conf. on Simulation of Semiconductor Processes and Devices*, San Diego, USA, September 2009, pp. 281–284.

[20] H. T. *et al*, "Advanced analysis and modeling of MOSFET characteristic fluctuation caused by layout variation," in *VLSI Technology Symposium*, Kyoto, Japan, June 2007, pp. 204 –205.

[21] "http://www.ciranova.com/."

[22] "http://www.cadence.com/."

[23] T. Massier, H. Graeb, and U. Schlichtmann., "The Sizing Rules Method for CMOS and Bipolar Analog Integrated Circuit Synthesis," vol. 27, no. 12, dec. 2008, pp. 2209 –2222.

[24] F. Javid, R. Iskander, and M.-M. Louërat, "Simulation-Based Hierarchical Sizing and Biasing of Analog Firm IPs," *IEEE International Behavioral Modeling and Simulation Conference*, pp. 43–48, Sep. 2009.