

Simulation-based low-level optimization tool for analog integrated circuits

Ginés Doménech–Asensi, José Alejandro López–Alcantud and Ramón Ruiz Merino

Departamento de Electrónica, Tecnología de Computadoras y Proyectos.
Universidad Politécnica de Cartagena, Dr. Fleming sn, 30201 Cartagena, Spain

ABSTRACT

In this paper, a tool based on free software to perform low level optimization on analog designs is presented. Nowadays, the use of design automation tools for microelectronic circuits design is extending from digital to analog circuits, due in part to the fact that although the analog part of a mixed signal ASIC takes only the 10% of the silicon area, it represents almost 90% of the whole design time. For analog circuits, design process can be divided in two major tasks: topology selection and device sizing. The tool here presented consists on a simulation based optimizer, which is used to perform automatic low level analog circuit sizing. The tool is composed of three modules: a layout generator, which includes a parasitic extractor, an analog circuit simulator and a circuit optimizer. The two first modules are respectively Magic and Spice from Berkeley, while the third one, the optimizer, has been developed to evaluate dc, ac, and transient sensitivity simulations performed by Spice and make corrections on the layout sizing. Optimization process starts with a certain topology and standard sized devices, which is then extracted by Magic and simulated by Spice. Performance is evaluated and a sizing correction is proposed. These simulations and corrections are done on an iterative loop until circuit performance reaches design parameters. The tool is demonstrated with an example of a simple analog subcircuit optimization, where parameters like silicon area or power dissipation are optimized, while the circuit keeps on design parameters

Keywords: CMOS analog integrated circuits, Electronic design automation, Circuit optimization, Spice

1. INTRODUCTION

In this paper, a tool based on free software to perform low level optimization on analog designs is presented. Nowadays, the use of design automation tools for microelectronic circuits design is extending from digital to analog circuits, due in part to the fact that although the analog part of a mixed signal ASIC takes only the 10% of the silicon area, it represents almost 90% of the whole design time.

Currently, there is a wide variety of EDA (Electronic Design Automation) tools, most of them developed specially for design, synthesis and verification of digital circuits. In the analog field, the number of EDA tools is still small in comparison with its digital counterparts, and the scope of each one is very different. A classification of analog EDA tools can be found in¹ and². According to these revisions, for analog circuits it is clear design process consist on two major steps: topology selection and analog circuit sizing. Topology selection is the more difficult to automate, since designer expertise is needed to achieve an optimal circuit architecture, and it is difficult to compile this knowledge in form of equations or tables good enough to be useful for several designs.

Circuit sizing is a more suitable task for automated design and optimization tools. This process can be addressed from two different strategies. The first and older one comprises the knowledge based techniques, where a set of already tested solutions are stored in the optimization software and are used when necessary. Some proposals in this field are IDAC³ or OASYS.⁴ However, this approach is not flexible and a lot of work is required before a good set of circuits can be included as solutions in the EDA tool. The second one relies on optimization based methods. These methods use numerical optimization techniques to solve an analog design

Further author information: (Send correspondence to Ginés Doménech Asensi.)

E-mail: gines.domenech@upct.es, Telephone: +34 968 326454

E-mail: ja.lopez@upct.es, Telephone: +34 968 326455

E-mail: ramon.ruiz@upct.es, Telephone: +34 968 326453

Copyright 2005 Society of Photo-Optical Instrumentation Engineers.

This paper was published in "Proceedings of SPIE" and is made available as an electronic reprint (preprint) with permission of SPIE. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

with several degrees of freedom while optimizing the performance of the circuit under a set of given specification constraints.

Optimization based methods for circuit sizing can be performed using equation based approaches, like OPASYN⁵ or by means of analog simulators like in DELIGHT.SPICE.⁶ This tool performs optimization in an iterative loop which include the use of Spice revised by the authors in order to include sensitivity analysis. Also, in⁷ the tool calls a plain Spice simulation in each iteration step, to perform operational amplifier optimization.

In general, for high level synthesis there have been several proposal for such EDA tools, not only with different optimization methods, but also each one designed for a specific purpose or design target, like analog to digital converters, filters, etc. In the field of low level synthesis tools the first approaches relied on procedural module generation⁸ in which the layout of the circuit was compiled in an EDA tool which generated the layout for current design parameters. However, as performance of analog circuits is deep impacted by parasitics derived from the layout, such parasitic capacitances and resistances or others, it was essential to take into account these effects in order to create compact layouts which satisfy all performance specifications. So, it was needed a new approach based on optimization, as for high level synthesis, as in ILAC⁹ or KOAN/ANAGRAM,¹⁰ optimization tools which borrowed main ideas from digital design.

According to previous review, our tool has been developed to perform analog circuit sizing, using a simulation based optimization. Although this proposal is valid both for high level synthesis and low level synthesis, we are going to work with low level synthesis, where analog simulator become essential for accurate estimation of circuit performance. At this hierarchy level, equation based optimization approaches are more inaccurate than for high level synthesis. It is essential to include parasitics extracted from layout in each iteration step. Our aim when developing this tool was to obtain an open environment which could be usable in academic environment for specific research applications. The tool is composed of three modules: a layout generator, which includes a parasitic extractor, an analog circuit simulator and a circuit optimizer. The two first modules are respectively Magic¹¹ and Spice3,^{12 13} from Berkeley, which are commonly used in academic environment. The third one, the optimizer, has been developed to evaluate dc, ac, and transient sensitivity simulations performed by Spice and make corrections on the layout sizing. Optimization process starts with a certain topology and standard sized devices, which is then extracted by Magic and simulated by Spice. In order to test individual modules interaction and general tool performance we have used a classic optimization algorithm. Its task is to evaluate circuit performance and propose a sizing correction in the layout. Simulations and corrections are done on an iterative loop until circuit performance reaches design parameters. This algorithm is described in further sections of the paper

This paper is organized as follows: in Sect. 1 we have presented a general description of our work. Structure of the proposed optimization tool is described in Sect. 2, while details of the algorithm are presented in Sect. 3. In Sect. 4 an example of a simple analog subcircuit optimization is done. Finally, Sect. 5 presents the conclusions of the tool proposed in the paper.

2. TOOL STRUCTURE

Figure 1 shows the structure of the optimization tool. The tool includes Magic and Spice3 from Berkeley, together with the extractor converter *ext2spice*, which are the modules darker in the figure. It is completed with a layout generator and an optimization module. Although Magic is itself a layout generator, here it is used only to display layouts created by other module and to perform parasitic extraction. Our aim when developing this tool was to obtain an open tool which could be usable in academic environment for specific research applications and for advance teaching of microelectronics design, reason for which we have employed well known and already tested tools like the above mentioned. These tools are available free and are commonly used in academic environments.

As we can see in the figure, main program controls all modules and performs an iterative loop. In this loop, the program runs Magic and Spice3 in background so, although source code of these tools is free, we have not needed to perform any modification on the original source codes. The main program as well as the rest of the modules have been developed in ANSI C for Linux, using Magic 7.2 and an adapted version of Spice3 for this operating system. The parasitics extractor is included as a Magic command and the converter from the extracted

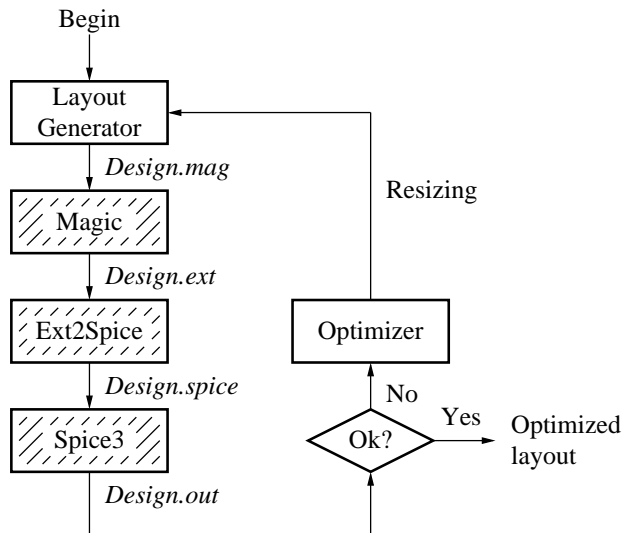


Figure 1. General structure of optimization tool.

circuit into the spice like file comes as an external utility in the same Magic package. Both the layout generator and the optimizer are easy to modify in order to include new optimization algorithms.

Optimization process starts with an initial layout defined in Magic format (*Design.mag*) which is created by the layout generator. This layout is extracted in order to obtain parasitics and the extracted file (*Design.ext*) is converted into an Spice (*Design.spice*) file using the *ext2spice* application. This is a raw Spice file which only contains instances of MOS devices and parasitics. The tool completes then the Spice description in order to obtain an useful file for simulation. These lines include device models for the technology used, sources for the type of analysis to perform and the specification of such analysis. Devices technology is provided by the user as a technology file which is read by the tool at the beginning of the optimization process. Also, the types of analysis are defined by the designer. Spice works with several types of analysis, from which dc, ac and sensitivity analysis are used in this optimization. User selects whether to use sensitivity analysis in combination with dc analysis or in combination with ac analysis depending on the goal specification and the design constraints to optimize. For some optimizations, both ac and dc can be used together with sensitivity analysis.

The spice like file obtained is the simulated using Spice3. The output (*Design.out*) is thus processed by the optimization tool. If performance meets specifications optimization finishes. In other case, optimization is performed, modifying certain design parameters in order to meet circuit specification. These modifications are sent to the layout generator which creates an updated Magic file and a new iteration step begins. In order to test individual modules interaction and general tool performance we have used a classic optimization algorithm. This optimization method is further detailed in section 3.

3. OPTIMIZATION METHOD

The general optimization method of an EDA tool and in general of most mathematical tools consist on the minimization of a cost function of design parameters subject to a set of constraints, such that:

$$\text{minimize} : O(p) \tag{1}$$

subject to:

$$c_i < C_i \tag{2}$$

where p are design parameters, c_i are the the constraints, such power dissipation, bandwidth, etc, and C_i are the maximum value for such constraint. In this way, the formulation is similar to a mathematical minimization problem.

Basing on definitions used in previous works¹⁴ design specification and design constraints can be divided in two categories: hard constrains, which include design specification and these constraints which imply physical realization (i.e. a transistor length positive) and soft ones which include the rest of design constraints. Following this approach, optimization is done in two steps, according to the technique proposed in¹⁴: a first iteration loop in which hard constraints are met and a second iteration loop in which soft constraints are improved, keeping design specifications.

We use a similar approach, taking advantage of the sensitivity analysis. In this way, at a certain iteration step, we know not only the error between desired performance and current specification, but also the direction in which changes in design parameter should be addressed. In the simulation based optimization, we rely on a combination of sensitivity analysis and ac or dc analysis. Let p_i be a design parameter, such W or L in a MOS transistor, e_i the error between the desired specification and the current performance at iteration step i , d_0 the specification goal and d_i the performance at step i . We have the following relationship:

$$e_i = d_0 - d_i \tag{3}$$

Thus, error e_i represents the increment desired in performance d_i to achieve specification goal d_0 . We can conclude that $\Delta d_i = e_i$. It is obvious that variation in performance d_i can be written as:

$$\Delta d_i = \Delta p_i \frac{\Delta d_i}{\Delta p_i} = \Delta p_i S_{p_i}^{d_i} \tag{4}$$

where $S_{p_i}^{d_i}$ is the sensitivity of specification d_i with respect to design parameter p_i . From (4) we obtain desired increment in p_i as:

$$\Delta p_i = \Delta d_i \frac{1}{S_{p_i}^{d_i}} \tag{5}$$

This equation represents desired increment in a design parameter in order to achieve specification d_0 . This expression implicitly assumes a linearity in the sensitivity function, which may not be correct, so Δp_i can be modulated by a factor $\delta < 1$ in order to give stability the iterative process. Equation (4) has been obtained for the particular case in which there is a single design parameter. A generalized expression is:

$$\Delta d_i = \sum \Delta p_1 S_{d_i}^{p_1} f_i \tag{6}$$

where the contribution of a single design parameter p_i , is modulated by a factor f_i . This factor f_i is computed for each design parameter p_i using sensitivity analysis. In order to find a suitable expression for f_i , let us assume constraint specifications are described in terms of minimization of their values. Figure 2 shows the sign of Δp_i and its influence in d_i and c_i .

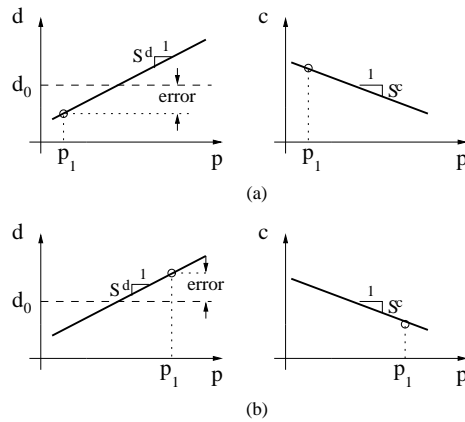


Figure 2. Inverter layout.

As we can see in Fig. 2(a) sensitivity of design specification d_1 with respect to parameter p_1 is positive, and at the current iteration step, the error between specification and current performance is also positive. If sensitivity of constraint with respect p_1 is negative then a variation of p_1 in order to correct Δd_1 improves also c_1 restriction. In Fig. 2(b) we have the same sensitivities, but this time, Δd_1 is negative. Thus, a change in p_1 in the direction to correct d_1 is not desirable as it increments c_1 . So, f_i must be calculated in order to find best circuit layout.

Table 1 shows the possible combinations of error e_i , and sensitivities $S_{p_i}^{d_i}$ and $S_{p_i}^{c_i}$.

Table 1. Combinations of sensitivities and errors in optimization function.

$S_{p_i}^{d_i}$	$S_{p_i}^{c_i}$	e_i	f
+	+	+	Bad
+	+	-	Ok
+	-	+	Ok
+	-	-	Bad
-	+	+	Ok
-	+	-	Bad
-	-	+	Bad
-	-	-	Ok

In this table, column f represents if the action performed by factor f is good or bad for minimizing constraint c_i . As we can see, only when the combinations of signs of $S_{p_i}^{d_i}$, $S_{p_i}^{c_i}$ and e_i is negative, the optimization steps improves results both for design performance d_i and for constraint c_i . In the rest of the cases, performance d_i is improved, while constraints c_i is not.

Let $S_{p_i}^{c_i}$ be the sensitivity of design constraint c_i with respect to design parameter p_i . We can find an expression in which f_i depends on the combination on sensitivity values, such that:

$$f_i = \frac{S_{c_i}^{p_i}}{\sum S_{c_i}^{p_i}} \quad (7)$$

Combination of Eqs. (7) and (6) accomplishes criteria defined for f in Tab. 1. So, variation in constraint c_i value is thus given by:

$$\Delta c_i = \sum \Delta p_i S_{c_i}^{p_i} \quad (8)$$

The procedure described above is used until specification goal is achieved. Once this step is over, it starts a second loop in which soft constraints are improved. Modification of design parameters will produce a design performance variation defined as:

$$\Delta d_i = \sum \Delta p_i S_{d_i}^{p_i} \quad (9)$$

However, in this second step specification goal should not be modified so $\Delta d_i = 0$, which yields:

$$\sum \Delta p_i S_{d_i}^{p_i} \quad (10)$$

4. EXAMPLE

We are going to use the method with an optimization example of a simple analog subcircuit. For this purpose we have used a CMOS inverter, using SCMOS technology. In this design, the objective function is to keep V_{out} while minimizing power dissipation and layout size. Power dissipation is proportional to current flowing in both

transistor channels. So, the objective is then to minimize I_D . As design parameters we have used W and L in both transistors.

Figure 3 shows the initial layout of a CMOS inverter, in which parameters W_1 , L_1 , W_2 and L_2 are our design parameters p_i . Applying Eqn.4 to increment in V_{out} at iteration step i (called V_i) as a function of parameter W_1

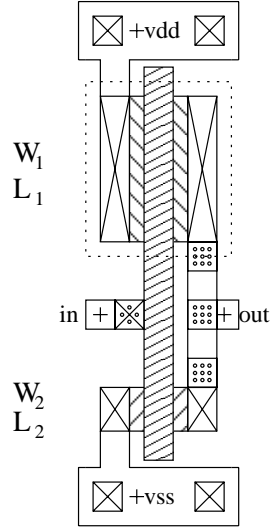


Figure 3. Inverter layout.

we have:

$$\Delta V_i = \Delta W_1 \frac{\Delta V_i}{\Delta W_1} = \Delta W_1 S_V^{W_1} \quad (11)$$

in the same way, for W_2 , L_1 and L_2 we obtain:

$$\Delta V_i = \Delta W_2 = \Delta W_2 S_V^{W_2} \quad (12)$$

$$\Delta V_i = \Delta L_1 = \Delta L_1 S_V^{L_1} \quad (13)$$

$$\Delta V_i = \Delta L_2 = \Delta L_2 S_V^{L_2} \quad (14)$$

If all four design parameters contribute in the same proportion to δV_i we have:

$$\Delta V_i = \frac{1}{4} \Delta W_1 S_V^{W_1} + \frac{1}{4} \Delta W_2 S_V^{W_2} + \frac{1}{4} \Delta L_1 S_V^{L_1} + \frac{1}{4} \Delta L_2 S_V^{L_2} \quad (15)$$

However, it is not probable that performance values have the same sensitivity to all design parameters. Using Eqn.(6) we obtain a generalized form of (15) as:

$$\Delta V_i = f_{W_1} \Delta W_1 S_V^{W_1} + f_{W_2} \Delta W_2 S_V^{W_2} + f_{L_1} \Delta L_1 S_V^{L_1} + f_{L_2} \Delta L_2 S_V^{L_2} \quad (16)$$

being the increments defined for individual parameters as:

$$\Delta W_1 = e_i \frac{1}{S_V^{W_1}} f_{W_1} \quad (17)$$

$$\Delta W_2 = e_i \frac{1}{S_V^{W_2}} f_{W_2} \quad (18)$$

$$\Delta L_1 = e_i \frac{1}{S_V^{L_1}} f_{L_1} \quad (19)$$

$$\Delta L_2 = e_i \frac{1}{S_V^{L_2}} f_{L_2} \quad (20)$$

In this example, suppose our specification is to obtain a CMOS inverter with $V_o = 3.0 V$ for bias point, minimizing I_D . For this circuit, we have the following initial device sizes: $W_1 = 200$, $L_1 = 20$, $W_2 = 40$ and $L_2 = 20$, expressed in terms of Magic square units. Table 2 shows the performance and sensitivity values obtained for the CMOS inverter initial layout. In this table we can see the initial error $e_i = \Delta V_i = 0.2096 V$ and the values

Table 2. Performance and sensitivities for initial layout.

Performance	Value
V_o	3.2096
I_o	190.9941 mA
Sensitivity	Value
$S_{L_1}^V$	-0.0065155411
$S_{W_1}^V$	0.0006515547
$S_{L_2}^V$	0.0065155402
$S_{W_2}^V$	-0.0032577734
$S_{L_1}^I$	0.0000000000
$S_{W_1}^I$	-0.0000000000
$S_{L_2}^I$	0.0000022979
$S_{W_2}^I$	-0.0000011489

of sensitivities used to compute f_i and then ΔW_i and ΔL_i . With this initial values iterative process starts until design specification $V_o = 3 V$ is reached. Final values for V_o and I_D are shown in table 3, where we can see still exist an small error between circuit specification and layout performance, where iterative process has entered in a loop where a better solution was not found. This is due this optimization algorithm can not detect if this is a local minimum or not.

Table 3. Performance for final layout.

Performance	Value
V_o	3.05704
I_o	171.8947 mA

Drain current has also been minimized in a proportion of 10.4%. Design parameters at the final iteration step are the following $W_1 = 198$, $L_1 = 21$, $W_2 = 37$ and $L_2 = 15$, in terms of Magic square units.

5. CONCLUSIONS

In this paper we have shown the structure of a CAD-EDA tool oriented to simulation driven low level analog design. This optimization is focused to device sizing in already placed and routed devices on an analog layout. The tool has been developed using already tested microelectronic analog design tools such Spice3 and Magic. These tools are been selected because they are commonly used in academic environments and constitute a known reference in analog design. The main objective of this work was to build a free tool which could be used in academic environment both for microelectronics advanced teaching and for academic research, as its architecture is open and several optimization algorithms can be developed and tested. In order to test tool performance and interaction between all modules which compose it we have used a simple optimization algorithm which in successive iteration steps, performs a size correction of CMOS devices in order to achieve a specification goal

subject to defined constraints. The algorithm includes sensitivity analysis for a driven adjustment in device sizing as it is essential when working at low hierarchy description level. The tool has been demonstrated with an example of a simple analog subcircuit optimization, where parameters like silicon area and power dissipation have been optimized, while the circuit kept on design constraints.

ACKNOWLEDGMENTS

This work has been supported by Fundación Séneca of Región de Murcia and Ministerio de Ciencia y Tecnología of Spain, under grants PB/63/FS/02 and TIC2003-09400-C04-02, respectively.

REFERENCES

1. G. G. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proceedings of the IEEE*, **88**, pp. 1825–1852, 2000.
2. S. Balkir, G. Dundar, and A. S. Ogreneci, *Analog VLSI design automation*, CRC Press, Washington D.C., 2003.
3. M. G. Degrauwe, O. Nys, J. Dijkstra, S. Bitz, B. Goffart, E. Vittoz, S. Cserveny, C. Meixenberg, G. Stappen, and H. J. Oguey, "IDAC: An interactive design tool for analog CMOS circuits," *IEEE Journal of Solid State Circuits*, **22**, pp. 1106–1115, 1987.
4. R. Harjani, R. Rutenbar, and L. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Transactions on Computer Aided Design*, **8**, pp. 1247–1265, 1989.
5. H. Koh, C. Sequin, and P. Gray, "OPASYN: A compiler for CMOS operational amplifiers," *IEEE Transactions on Computer Aided Design*, **9**, pp. 113–125, 1990.
6. W. Nye, D. C. Riley, A. Sangiovanni-Vincentelli, and A. L. Tits, "DELIGHT.SPICE: An optimization-based system for the design of integrated circuits," *IEEE Transactions on Computer-Aided Design*, **7**, pp. 501–519, 1988.
7. F. Medeiro, "An statistical optimization-based approach for automated sizing of analog cells," in *Proceedings of the ACM/IEEE, International Conference on Computer-Aided Design (ICCAD)*,
8. J. Kuhn, "Analog module generation for silicon compilation," *VLSI system design*, , 1987.
9. J. Rijmenants, "ILAC: An automated layout tool for analog CMOS circuits," *IEEE Journal of Solid State Circuits*, **24**, pp. 417–425, 1989.
10. J. Cohn, D. Garrod, R. Rutenbar, and L. Carley, *Analog device level layout generation*, Kluwer, Norwell, MA, USA, 1994.
11. J. K. Ousterhout, G. T. Hamachi, R. N. Mayo, W. S. Scott, and G. S. Taylor, "Magic: a VLSI layout system," in *Proceedings of the 21st Design Automation Conference*, pp. 152–159, 1984.
12. L. W. Nagel, *SPICE2: A computer program to simulate semiconductor circuits*, Ph.D. Dissertation, Univ. of California, Berkeley, 1975.
13. B. Johnson, T. Quarles, A. R. Newton, D. O. Pederson, and A. Sangiovanni-Vicentelli, *SPICE3 Version 3e User's manual*, Dept. of Electrical Engineering and Computer Sciences, Univ. of California, Berkeley, April 1991.
14. W. Nye and A. L. Tits, "An application-oriented, optimization based methodology for interactive design of engineering systems," *International Journal of Control*, **43**, pp. 1693–1721, 1986.