GENOM-POF: Multi-Objective Evolutionary Synthesis of Analog ICs with Corners Validation

Nuno Lourenço
Instituto de Telecomunicações
Instituto Superior Técnico
IST – Torre Norte, AV. Rovisco Pais, 1
1049-001 Lisboa, Portugal
nlourenco@lx.it.pt

Nuno Horta
Instituto de Telecomunicações
Instituto Superior Técnico
IST – Torre Norte, AV. Rovisco Pais, 1
1049-001 Lisboa, Portugal
nuno.horta@lx.it.pt

ABSTRACT
In this paper, a multi-objective design methodology and tool for automatic analog IC synthesis, which takes into account the effects of process variations, is presented. By varying the technological and environmental parameters, the robustness of the solutions is enhanced. The automatic analog IC sizing tool, GENOM-POF, was implemented to demonstrate the methodology and to verify the effects of corner cases on the Pareto optimal front (POF). The impacts of NSGA-II parameters when applied to analog circuit sizing were investigated, and three different design strategies were tested in a benchmark circuit, showing the effectiveness of multi-objective design of analog cells.

Categories and Subject Descriptors
I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems – *industrial automation,* J.2 [**Computer Applications**]: Physical Sciences and Engineering – *electronics.*

General Terms
Algorithms, Performance, Design, Reliability

Keywords
Multi-Objective Optimization, Analog IC Sizing, Microelectronics, Electronic Design Automation, Computer Aided Design

1. INTRODUCTION
In the last decades, Very Large Scale Integration (VLSI) technologies have been widely improved, allowing the proliferation of consumer electronics and enabling the growth of Integrated Circuit (IC) market from $10 billion in 1980 to more than $300 billion in 2013 (according to IC Insights Inc) [24]. At the same time, the need of new functionalities, longer battery times, smaller (thinner) devices, more power efficiency, less production and integration costs and less design cost, makes the design of electronic systems a truly challenging task. IC designers are building systems that are increasingly more complex and the integration in modern systems is extremely high. In the System on Chip (SoC) age it is common to find devices where the whole system is integrated in a single chip.

GECCO'12, July 7–11, 2012, Philadelphia, Pennsylvania, USA.

The complexity of electronic systems, the extremely competitive markets, and the strict time-to-market impose the use of Computer Aided Design (CAD) tools to support the design process. In digital IC design, several Electronic Design Automation (EDA) tools and design methodologies are available that help the designers keeping up with the new capabilities offered by the technology. Currently almost all low-level phases of the process are automated. The level of automation is far from the push-button stage, but is keeping up reasonably well with the complexity supported by the technology. On the other hand, analog design automation (ADA) tools are not keeping up with new challenges created by technological evolution [12, 15]. Due to the lack of automation, designers keep exploring the solution space manually. This method causes long design times, and allied to the non-reusable nature of analog IC, making analog IC design a cumbersome task. The International Technology Roadmap for Semiconductors 2009 report [15] showed the differences between analog and digital design automation. This difference in automation is because analog design in general is less systematic, more heuristic and knowledge intensive than the digital counterpart, and becomes critic when digital and analog circuits are integrated together.

In this paper a methodology and tool for automatic analog IC synthesis, GENOM-POF, is presented. GENOM-POF stems from GENOM [2-4], a former single objective optimizer enhanced by an SVM feasibility model. The main novelty of this work is the inclusion of multi-objective optimization in a robust synthesis approach to analog IC design.

This paper is organized as follows: in section 2 an overview of analog IC design with special emphasis to analog IC sizing is presented; section 3 highlights previous work in automatic analog IC synthesis; section 4 explains the architecture of GENOM-POF; section 5 presents case studies; and finally, in section 6 some conclusions are drawn and future work proposed.

2. ANALOG IC DESIGN: OVERVIEW
In order to locate analog IC sizing, a brief presentation of a typical analog IC design flow is shown, and the analog IC sizing task is described.

2.1 Design Flow
A commonly well accepted design flow for analog and mixed-signal ICs is depicted in Figure 1. It was proposed by Gielen and Rutenbar in [12] and consists of a series of top-down topology selection and specifications translation steps and bottom-up layout generation and extraction steps including several verification stages along the way.

On the top-down path, topology selection is the process where a set of blocks and connections between them are defined in order to implement the input specifications of the current hierarchy level. In specification translation, higher-level specifications are translated into specifications for each of the blocks. Block specifications can be the gain or bandwidth of an amplifier, or the transistors' sizes (of the circuit implementing that amplifier), depending of the models used in the abstraction level in question. The sizing is then verified to ensure it performs according to specifications. The specifications for each block are then passed to the next level of hierarchy, and the process is repeated until the layout of the inner most block is done.

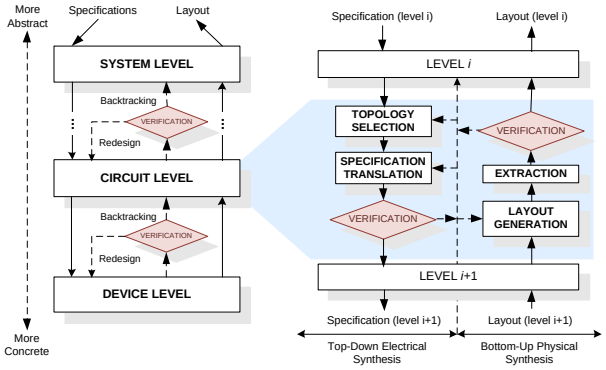


Figure 1. Hierarchical level and design tasks of analog design flow architecture

At this point, the bottom-up flow is executed. First, the layout of the current level is generated using the layouts from the previous level. Then, it is extracted to a model suitable for verification, and the specifications are validated. When the top-most level verification is complete, the system is designed.

It is important to note that any verification stage throughout the design cycle may detect potential problems, with the design failing to meet the requirements. In that case, backtracking or redesign will be needed.

2.2 Circuit Sizing

GENOM-POF addresses the problem of automatic specification translation at circuit level, also known as circuit sizing, where from the set of specifications, the designer finds out the sizes for the components, e.g., widths and lengths of the transistors, resistors, capacitors, etc..

In the industry, this task is commonly done manually. The designers start by finding an approximate solution using simplified analytical expressions, and then, iteratively, adjust the solution until it meets all specifications. The verification is done using electrical simulations, which sometimes can be very time consuming.

One of the critical problems in analog IC design is the process variability, i.e., devices designed to be equal are different after production. This phenomenon affects devices in different chips but also devices within the same chip, and must be solved by robust circuit design with several compensatory techniques. To verify if the design is robust, i.e., the vast majority of the fabricated circuits will work according to specifications, special analysis techniques are employed. The most common techniques for analog design centering are Monte Carlo Simulation and Corner Analysis. Monte Carlo simulation executes many simulations applying random variations to circuit's and process' parameters, making a stochastic sampling of the behaviors of the circuit in real world conditions. Corner Analysis is a worst-case approach where the circuit is simulated over multiple combinations of process parameters variations, e.g., power supply, temperature, etc.. Figure 2 illustrates 27 corners cases obtained by considering 3 values for power supply, operating temperature and process parameters.

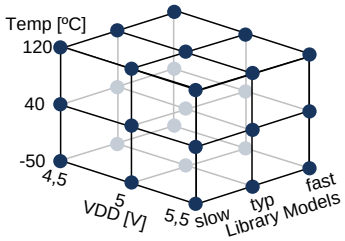
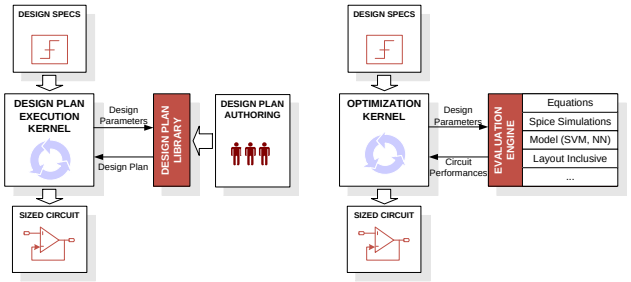


Figure 2. Corner cases

The designer experience and knowledge are of the utmost importance in analog IC design, as they allow simplifications that speed up the design process, without compromising the quality of the solution. Despite the fact that the design is still mostly handmade, the scientific community has been, in the past two decades, developing new tools and techniques to automate analog IC sizing. The next section presents an overview of those works.

3. PREVIOUS WORK

Historically, the tools for automated circuit sizing are classified as knowledge-based or optimization based, illustrated in Figure 3, this classification is based on the fundamental techniques used to address the problem.



(a) knowledge-based (b) optimization-based

Figure 3. Automatic specification translation

The next sections provide an overview of these two classes of tools.

3.1 Knowledge-based sizing

Early strategies tried to systematize the design by using a design plan derived from expert knowledge. In these methods, a pre-designed plan is built with design equations and a design strategy that produce the component sizes that meet the performance requirements. In IDAC [8], the designer expertise is captured in a design plan where all equations are explicitly solved during the definition of the plan. A different approach is found in BLADES [10], as it captures the designer's knowledge in expert systems using artificial intelligence techniques.

The knowledge-based approach was applied with moderate success. The main advantage of this approach is the short execution time. On the other hand, deriving the design plan is hard and time-consuming, the design plan requires constant

maintenance in order to keep it up to date with technological evolution, and the results are not optimal, suitable only as a first-cut-design.

3.2 Optimization-based sizing

Aiming for optimality, the next generations of sizing tools apply optimization techniques to analog IC sizing. Based on the evaluation techniques employed, the optimization-based sizing tools can be further classified into three main subclasses: equation-based, numerical-simulation-based, and numerical-model-based.

3.2.1 Equation-based

The equation-based methods use analytic design equations to evaluate the circuit performance. In GPCAD [13] a posynomial circuit model is optimized using Geometrical Programming (GP), the execution time is in the order of few seconds, but the general application of posynomial models is difficult and the time to derive the model for new circuits is still high. To reduce the long time dispended in model development, automatic techniques were proposed (Gielen et. al. in [11] provide a good overview on symbolic analysis applied to analog ICs). However, some design characteristics are still not easy to extract from analytical expressions with satisfactory accuracy automatically. Kuo-Hsuan et. al. [18] revisited the posynomial modeling recently, surpassing the accuracy issue by introducing an additional generation step, where local optimization using simulated annealing and a circuit simulator is performed.

The equation-based methods' strong point is the short evaluation time, making them, like the knowledge-based approaches, extremely suited to derive first-cut designs. The main drawback is that not all design characteristics can be easily captured by analytic equations, in addition, the approximations introduced in the equations yield low accuracy designs especially for complex circuits.

3.2.2 Numerical-simulation-based

Numerical-simulation-based sizing techniques use a circuit simulator to evaluate the circuit's performance. Generality and easy-and-accurate model are the strong points of simulation-based techniques. However these techniques exhibit large execution times for complex circuits.

Being the high execution time the weaker point of this approach, some techniques had been proposed to cope with it. Kuo-Hsuan et. al. [18] used equation-based techniques to derive an approximate initial solution. Cheng et al. [5] instead of solving the circuit by finding transistor sizes, solved it by finding the bias of the transistors first, and then , the transistor sizes are derived from the bias point using electric simulation. In MAELSTROM and ANACONDA [27] the evaluation time is reduced by a parallel mechanism that shares the evaluation load among multiple computers.

3.2.3 Numerical-model-based

The numerical-model-based tools use macro models, like neural-networks or support vector machines (SVM), e.g., to evaluate the circuit's performance. Given the high execution times caused by the use of electrical simulation inside the optimization loop, especially at systems level, learning techniques (like neural networks or SVMs) are used to create accurate models of the circuits suitable to replace the simulator. Usually these models are automatically generated using an electric simulator to evaluate the performance of the training set. Alpaydin et. al. [1] use a neural-fuzzy model combined with an evolutionary optimization strategy where some of the AC performance metrics are computed using an equation-based approach. Barros et. al. [4] present a cell-level synthesis and optimization approach based on SVMs and evolutionary strategies, which are used to dynamically model performance space and identify the feasible design regions, while at the same time the evolutionary techniques are looking for the global optimum.

A different approach is the use of Pareto optimal fronts (POFs) to explore the tradeoff during synthesis. Instead of using a model for the circuits, the non-dominated solutions are generated and the suitable solution is selected from the already sized solutions. [9] use POFs hierarchically to perform system level sizing. The POF-based-design execution time is large if the setup time is considered, however with the correct models, they can be generated in a context free manner [28] making then suitable for reuse.

3.2.4 Simultaneous topology selection and sizing

In MINLP [23], DARWIN [17] and SEAS [26] device sizing and topology selection are done simultaneously. This design mechanism uses a template library. This template specifies the topology in terms of blocks, each one with possible different alternatives. The selection is done using optimization techniques (linear programming for the first and genetic/evolutionary programming for the others) that explore the search space defined by the templates. These methods are more reliable than other topology selection techniques, as they treat the problem in a unified manner. The computation time, however, is extremely high. MOJITO [25] takes a similar approach but use hierarchical templates with variable structure that allow a deeper exploration of the design space while retaining some control of the architectural results obtained.

A different approach to topology selection is to generate new topologies rather than exploring the options available in a library. Koza [16], Lohn [19], Sripramong [30], Shoou-Jin [29] and more recently Hongying [14] presented a design methodology able to create new topologies by exploring the immense potential starting from a low level of abstraction. Small elementary blocks are connected to each other to form a new topology. Various fundamental entities can be applied, such as, single transistors, elementary building blocks or node connections. However, this generation is only possible at circuit level, as the computation time becomes unmanageable with the increase in the number of components. Another issue with bottom-up generation is that designers are suspicious of the generated structures as they sometimes differ too much from well-known trusted analog circuits [21]. Figure 4 shows the panorama of analog circuit synthesis contributes.

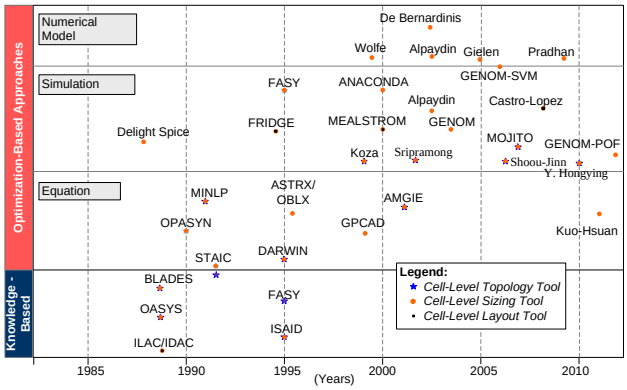


Figure 4. Overview of analog design automation tools

4. ARCHITECTURE

The proposed tool for analog circuit synthesis, GENOM-POF, is based on the elitist multi-objective evolutionary optimization kernel NSGA-II [7], and uses the industrial grade simulator HSPICE® to evaluate the performance of the design. GENOM-POF targets the design of robust circuits, by allowing the consideration of corner cases during optimization.

Figure 5 shows the architecture of the developed tool. GENOM-POF inputs the circuit netlist and testbench, the definition of the optimization variables, design constraints and objectives, and the corners cases. The circuit is then modeled as an optimization problem suitable to be optimized by the NSGA-II kernel. The output is a family of Pareto optimal sized circuits, representing the feasible tradeoffs between the different optimization objectives. The outputs are then used by the in house layout generator, LAYGEN II [20, 22], to automatically generate the physical implementation of the circuit.

The next subsections provide the details of the architecture using a simple circuit to illustrate the descriptions.

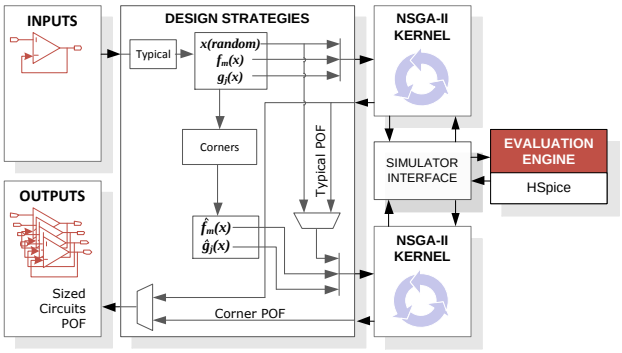
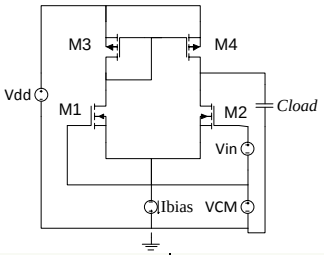


Figure 5. GENOM-POF architecture

4.1 Inputs

The inputs from the designer are the circuit and test-benches in the form of HSPICE® netlist(s). The netlist(s) must have the optimization variables as parameters, and must include means to measure the circuit's performance, the corner's parameter variations are also included in the netlist. Figure 6 shows a differential amplifier with the test-bench schematic and exerts of the corresponding netlist.




```
* TESTBENCH
Vdd 6 0 5V
V- 2 0 2.5V
…
* AC analysis
.ac dec 200 1 1000Meg …
…
* CIRCUIT
M1 5 3 1 0 nmos L=11 W=w1
M2 4 7 1 0 nmos L=11 W=w1
M3 5 5 6 6 pmos L=12 W=w2
M4 4 5 6 6 pmos L=12 W=w2
```

```
* MEASURES OF
* PERFORMANCE
.measure ac 'gain_dc' …
.measure ac 'gbw' …
…
* OVERDRIVES
.measure ac vov_m1 …
…
* MARGINS
.measure ac delta_m1 …
…
* TEMPERATURE CORNERS
.temp -40 +50 +120
```

Figure 6. Example circuit and test-bench netlist

In addition, the designer defines ranges for the optimization variables, design constraints, and optimization objectives. Table 1 shows these definitions for the circuit in Figure 6.

Table 1. Range, objectives and design constraints example

Variables	`w1, w2, l1, l2, ib`

Ranges:	`1.0e-6 <= w1 <= 500.0e-6`
	`1.0e-6 <= w2 <= 500.0e-6`
	`0.35e-6 <= l1 <= 15.0e-6`
	`0.35e-6 <= l2 <= 15.0e-6`
	`30.0e-6 <= ib <= 400.0e-6`

Objectives:	`max(gain_dc)` `min(rmspower)`

Constraints:	`gbw .. >= 35e6`
	`pm ... >= 65`
	`pm ... <= 90`
	`vov_m1 >= 50e-3`
	`vov_m2 >= 50e-3`
	`vov_m3 >= 100e-3`
	`vov_m4 >= 100e-3`
	`vov_m1 <= 200e-3`
	`vov_m2 <= 200e-3`
	`vov_m3 <= 300e-3`
	`vov_m4 <= 300e-3`
	`delta_m1 >= 50e-3`
	`delta_m2 >= 50e-3`
	`delta_m3 >= 50e-3`
	`delta_m4 >= 50e-3`

4.2 Optimization Kernel

The optimization engine used in GENOM-POF is NSGA-II modified to interface with HSPICE® which is used to evaluate the individual objective and constraint functions.

The NSGA-II was selected over SPEA and other multi-objective evolutionary algorithms because of the good characteristics of the output Pareto [7]. The option of using HSPICE® to evaluate the circuit's performance was due to the accuracy of the results and the availability of models for the devices provided by the foundries, despite the higher execution times of an electrical simulation.

The multi-objective optimization kernel module was designed to solve the problem:

$$\begin{aligned} &find\ x\ that\ minimizes\ f_m(x) && m = 1,2,\dots M \\ &subject\ to\ g_j(x) \geq 0 && j = 1,2,\dots .J \\ &x_i^L \leq x_i \leq x_i^U && i = 1,2,\dots .N \end{aligned} \tag{3.1}$$

where, x is a vector of N optimization variables, $g_j(x)$ one of the j constraints and $f_m(x)$ one of the M objective functions.

Two convergence measures were introduced to control the stop condition of the algorithm based on the convergence rate. One measures the ratio between the number of points in the POF and the population size, and the other, measures the ratio between the area under the POF and the ranges of the objectives. The expressions for these two measures are shown in eq. 3.2 and 3.3, respectively:

$$h_0(g) = \frac{\#pof}{\#pop} \tag{3.2}$$

where *#pof* is the number of points in the POF and *#pop* is the size of the population:

$$h_1^G(g) = \frac{2}{M(M-1)} \times \sum_{i=0}^{M} \sum_{j=i+1}^{M} \frac{area_G(i,j)}{\Delta_G(f_i(x)) \times \Delta_G(f_j(x))} \tag{3.3}$$

where $area_G(i,j)$ is the area under the 2D curve formed by the points $\{f_i(x), f_j(x)\}$ until the minimum values of

$f_i(x)$ and $f_j(x)$ found until generation G and $\Delta_G(f_i(x))$ is the range of $f_i(x)$ found from generation 0 to generation G, i.e., for each new generation, G, $h_1^G(g)$ needs to be calculated for all $g\epsilon\{1..G\}$.

The stop condition is:

$$\frac{1}{10}\sum_{i=0}^{9} h_0^G(G-i) > H_0 \quad \text{and} \quad h_1^G(G) - \frac{1}{10}\sum_{i=1}^{10} h_1^G(G-i) < H_1 \tag{3.4}$$

where the values for H_0 and H_1 are 0.95 and 10^{-5} by default.

Except for these modifications, the optimization kernel was implemented as in [7], using simulated binary crossover and mutation operators [6], tournament selection, and constrained based dominance check.

4.3 Design Strategies

GENOM-POF supports three design strategies Typical, Corners, and Typical plus Corners. The next subsections describe each of the strategies.

4.3.1 Typical (T)

As the name states, in this strategy the circuit is evaluated using only typical conditions, this strategy is faster, and even though the output does not consider the limitations imposed by the corners it is useful for design tradeoffs analysis.

First the design problem is described as an optimization problem, then the NSGA-II optimization kernel can be executed. The design objectives being minimized are used directly as one of the $f_m(x)$, and the ones being maximized are multiplied by -1. The design constraints are normalized and multiplied by -1 if necessary according to eq. 3.5:

$$g_j(x) = \begin{cases} \frac{p_j}{P_j} - 1 \; when\ the\ design\ constraint\ is\ p_j \geq P_j \\ -\left(\frac{p_j}{P_j} - 1\right) \; when\ the\ design\ constraint\ is\ p_j \leq P_j \end{cases} \tag{3.5}$$

where, p_j is the measured performance characteristic, and P_j is the corresponding acceptable limit.

Table 2 illustrates the objective and constraint functions for the circuit in Figure 6 using the design specifications in Table 1.

Table 2. $f_m(x)$ and $g_j(x)$ example

Objectives:	$f_0(x)$ = -gain_dc $f_1(x)$ = rmspower
Constraints:	$g_0(x) = \frac{gbw}{35\times10^6} - 1$ $g_1(x) = \frac{pm}{65} - 1$ $g_2(x) = 1 - \frac{pm}{90}$ $g_3(x) = \frac{\text{vov\_m1}}{50\times10^{-3}} - 1$ $g_4(x) = \frac{\text{vov\_m2}}{50\times10^{-3}} - 1$ .... $g_{15}(x) = \frac{\text{delta\_m4}}{50\times10^{-3}} - 1$

4.3.2 Corners (C)

In the Corners strategy, the design is optimized from the beginning using all the corners, i.e., for each evaluation the circuit is simulated once for each corner case, this makes it the slower strategy, but the output circuits are feasible in all tested corner conditions.

To handle the multiple corners, the objective and constraint functions are modified using eq. 3.6, 3.7 and 3.8:

$$\hat{f}_m(x) = \max_{c=1,2..C}\left(f_m^c(x)\right) \tag{3.6}$$

$$\hat{g}_j(x) = \sum_{c=1}^{C} c_j^c(x) \tag{3.7}$$

$$c_j^c(x) = \begin{cases} 0 & if\ g_j^c(x) \geq 0 \\ g_j^c(x) & if\ g_j^c(x) < 0 \end{cases} \tag{3.8}$$

where, C in the number of corners and $f_m^c(x)$ and $g_j^c(x)$ are respectively the objective $f_m(x)$ and the constraint $g_j(x)$, as defined for the typical case evaluated in corner case c.

4.3.3 Typical plus Corners (TC)

In this strategy, typical optimization is done first, and then, the typical POF is used as starting point for corner optimization. This is a tradeoff between the execution time and robustness of the solution, and the reduction of the genetic information (localization of the search) imposed by the use of the typical POF as starting point of the corner optimization.

4.4 Outputs

The output is the family of sized circuits that represent the possible tradeoffs between the objectives being optimized. Figure 7 shows the frequency response for the family of solutions obtained for the circuit from Figure 6.

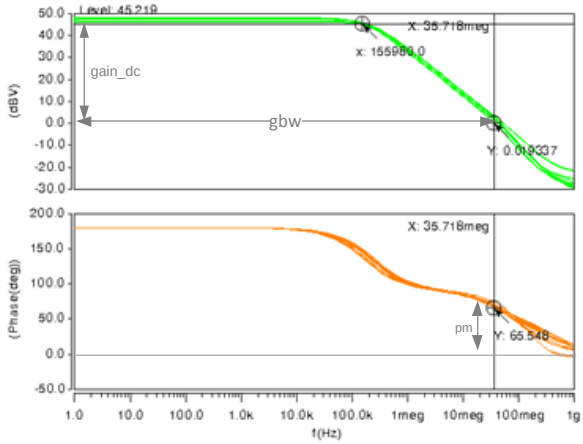


Figure 7. Frequency response for the output POF

5. CASE STUDY

The GENOM-POF tool was used for typical and corner design of a single ended folded cascade amplifier. The circuit is described in section 5.1, the obtained results are presented in sections 5.2 and 5.3.

5.1 Circuit Description

In Figure 8 the circuit schematics is shown, and the ranges, objectives and constraints are listed in Table 3. The problem has 15 real variables, 2 objectives and 19 constraints.

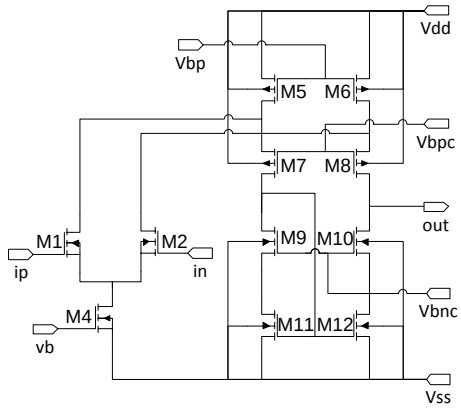


Figure 8. Single-ended folded cascade amplifier schematic

Table 3. Range, objectives and constraints

Variables:	cn, cp, l1, l4, l5, l7, l9, l11, ib, w1, w4, w5, w7, w9, w11
Ranges:	0.18e-6 <= l* <= 5.0e-6 0.24e-6 <= w* <= 200.0e-6 -0.4 <= cn <= 0.0 0.0 <= cp <= 0.4 30.0e-6 <= ib <= 400.0e-6
Objectives:	min(area) min(power)
Constraints:	gb >= 1.2e7 a0 >= 70 55 <= pm <= 90 sr >= 1e7 ov_m(*) >= 30e-3 d_m(*) >= 1.2 osp >= 0.5 osn <= -0.5 (*) the constraints apply to: M1, M4, M5, M7, M9 and M11

In addition, 9 corner cases were defined using the combination of technology models (typical, fast and slow) and temperature values (-40ºC, 50ºC, 120ºC). All the presented results are for *UMC 0,18µm* technology and include only feasible solutions.

5.2 Typical case optimization

In order to understand the behavior of the circuit and algorithm some tests were executed to see the variation of the result with the algorithm parameters (population size, mutation and crossover probability). The algorithm convergence measures were also analyzed. Figure 9 shows the evolution of convergence measures with the number of generations in a 32-element population.

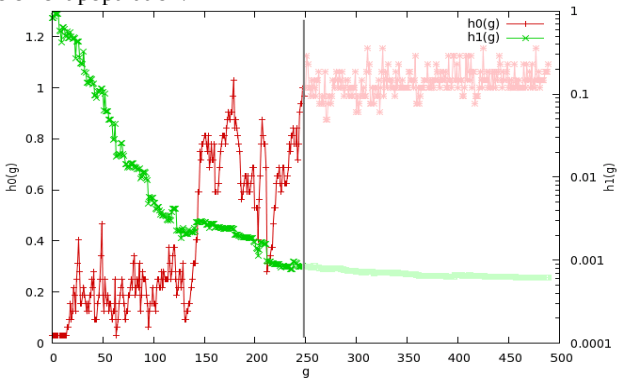


Figure 9. Evolution of the h_0 and h_1 with the generations

The vertical line marks the stop condition and the lighter points were obtain by letting the algorithm run until the max generation limit. Figure 10 shows the evolution of the POF for the same run, lighter POFs were obtained after the convergence criteria and as shown do not represent considerable improvements.

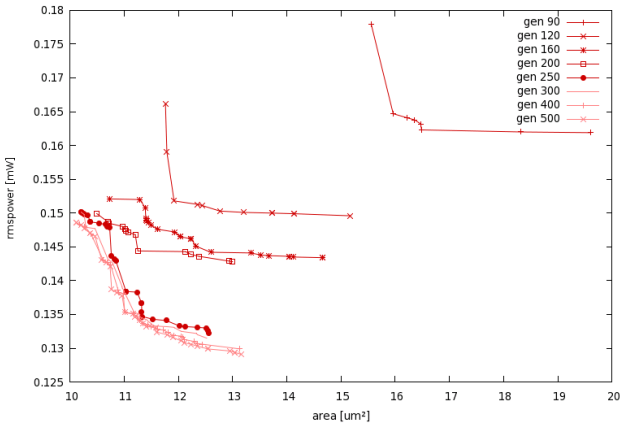


Figure 10. Evolution of the POF with the generations

Figure 11 shows the POFs obtained by varying the population size. It is noticeable that for population size larger than 80 the improvement in the POF is negligible.

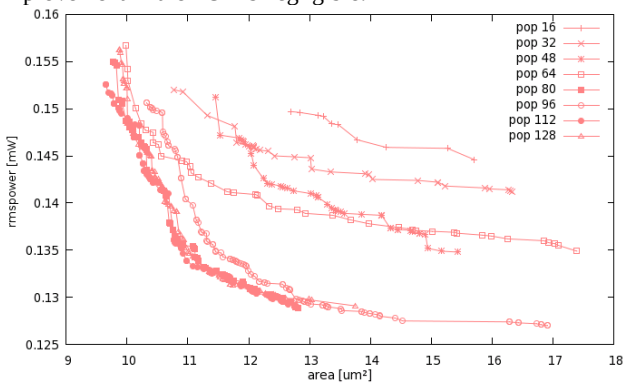


Figure 11. Effects of population size in the Circuit POF

Figure 12 and Figure 13 show the effects of varying the crossover and mutation rates respectively. The % of crossover did not affect the POF significantly after 50%, nevertheless larger crossovers (>90%) yield better results. The mutation rate around 15% (green and yellow POFs) presented better results.

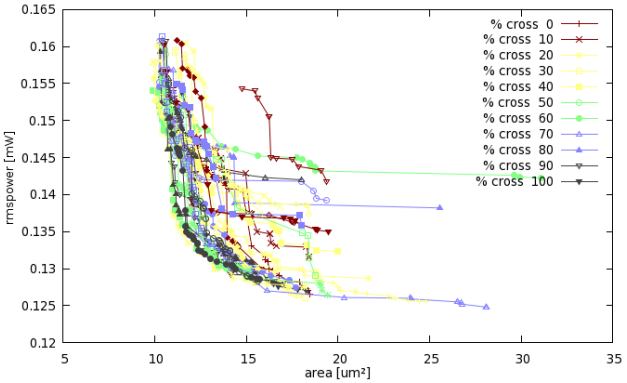


Figure 12. POFs for various crossover % (32 elem, 200 gen)

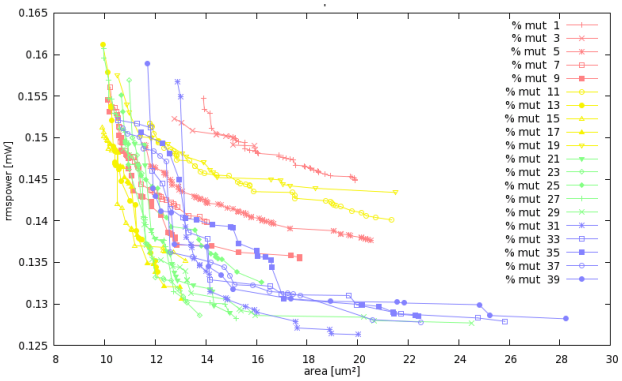


Figure 13. POFs for various mutation % (32 elem, 200 gen)

5.3 Corner and Typical plus Corners case optimizations

In order to design robust circuits, process variations must be considered. In this section the performance of GENOM-POF while handling circuit optimization with corners, was analyzed.

Figure 14 shows the POFs evolution for both C (in blue) and TC (in green and red) strategies (see sections 4.3.2 and 4.3.3 for strategy definitions). Both runs were executed with a population of 32 individuals and 100 generations. Table 4 summarizes the output. For the same number of generations, C is slower; in addition, TC provides better results. By starting the corner optimization from the already optimized typical POF, it is easier to fulfill the additional constraints imposed by the corners.

Table 4. Corner and Typical plus Corner summary

Strategy	Corner (C)			Typical plus Corner (TC)		
	Up	Middle	Down	Up	Middle	Down
Area [um2]	3.11e+01	3.50e+01	3.68e+01	1.99e+01	2.08e+01	2.25e+01
Power [mW]	3.57e-01	2.06e-01	2.06e-01	2.00e-01	1.76e-01	1.75e-01
Time [s]	572 (570 in simulator)			227(226 in simulator)		

Figure 15 shows the POFs and execution time that were obtained by running the 3 strategies T, C and TC until convergence, with a population of 80 elements, crossover rate of 90%, 10% of mutation rate and H_0 equals to 0,7. As expected, the POF obtained using T was found faster, and dominates the others (because it has fewer constraints). TC strategy was faster than C and provided better results in a region of the POF, however it did not dominate the POF obtained using C completely.

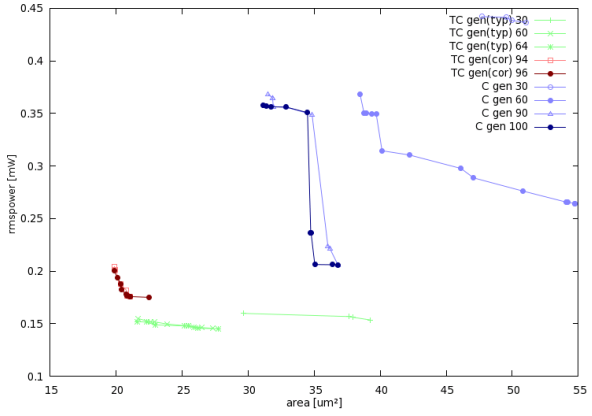


Figure 14. POF for corner optimization using C and TC

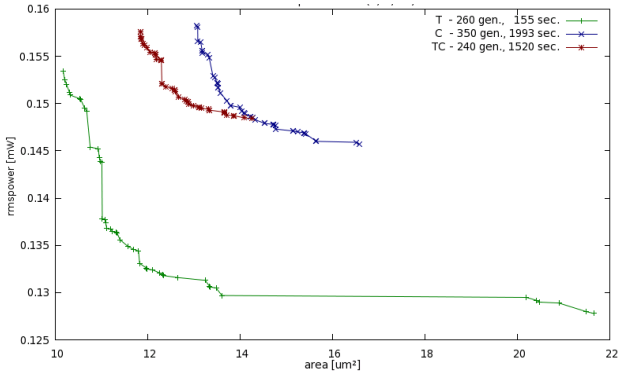


Figure 15. POF for T, C and TC after convergence

6. CONCLUSIONS

The proposed methodology and tool, GENOM-POF, were used to successfully design well known analog circuits, taking into account robustness consideration by the inclusion of corner cases. Moreover, the multi-objective nature of the IC design synthesis makes it well suited for automatic design using multi-objective optimization strategies. The usefulness of GENOM-POF to designers was shown using different design strategies. First, using the Typical (T) design strategy, the designer explores several design tradeoffs in a matter of minutes, which is useful for system level design. Then, using the C or TC strategies the designer can obtain a family of optimum robust circuits that comply with the specification in all corner cases considered. Moreover, GENOM-POF was also used to analyze the impact of the NSGA-II parameters when applied to analog IC sizing. Finally, the proposed methodology and tool are general once they have no dependence on the circuit or technology or, even, in the number of defined objectives.

7. ACKNOWLEDGMENTS

This work was supported in part by the Fundação para a Ciência e Tecnologia (Grant FCT-DFRH-SFRH/BD/72698/2010) and by the Instituto de Telecomunicações (Research project AIDA - IT/LA/1112/2011)

8. REFERENCES

[1] Alpaydin, G., Balkir, S. and Dundar, G. 2003. An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. *IEEE T Evolut Comput.* 7, 3 (May. 2003), 240-252. DOI= http://dx.doi.org/10.1109/TEVC.2003.808914

[2] Barros, M., Guilherme, J. and Horta, N. 2007. GA-SVM feasibility model and optimization kernel applied to analog IC design automation. In *Proceedings of the ACM Great Lakes symposium on VLSI* (Stresa-Lago Maggiore, Italy, March 11 - 13, 2007), 469-472. DOI= http://dx.doi.org/10.1145/1228784.1228895

[3] Barros, M., Guilherme, J. and Horta, N. 2010. Analog circuits optimization based on evolutionary computation techniques. *Integr VLSI J.* 43, 1 (Jan. 2010), 136-155. DOI= http://dx.doi.org/10.1016/j.vlsi.2009.09.001

[4] Barros, M. F. M., Guilherme, J. M. C. and Horta, N. C. G. 2010. *Analog circuits and systems optimization based on evolutionary computation techniques*. Springer, Berlin.

[5] Cheng-Wu, L., Pin-Dai, S., Ya-Ting, S. and Soon-Jyh, C. 2009. A bias-driven approach for automated design of operational amplifiers. In *Proceedings of the International Symposium on VLSI Design, Automation and Test*

(Hsinchu, Taiwan, April 28 - 30, 2009), 118-121. DOI= http://dx.doi.org/10.1109/VDAT.2009.5158109

[6] Deb, K. and Agrawal, R. B. 1995. Simulated binary crossover for continuous search space. *Complex Systems*. 9, 2 (1995), 115-148.

[7] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T Evolut Comput*. 6, 2 (Apr. 2002), 182-197. DOI= http://dx.doi.org/10.1109/4235.996017

[8] Degrauwe, M. G. R., Nys, O., Dijkstra, E., Rijmenants, J., Bitz, S., Goffart, B. L. A. G., Vittoz, E. A., Cserveny, S., Meixenberger, C., Stappen, G. v. d. and Oguey, H. J. 1987. IDAC: an interactive design tool for analog CMOS circuits. *IEEE J Solid-St Circ*. 22, 6 (Dec. 1987), 1106-1116. DOI= http://dx.doi.org/10.1109/JSSC.1987.1052861

[9] Deniz, E. and Dundar, G. 2010. Hierarchical performance estimation of analog blocks using Pareto Fronts. In *Proceedings of the Conference on Ph.D. Research in Microelectronics and Electronics* (Berlin, Germany, July 18 - 21 2010), 1-4.

[10] El-Turky, F. and Perry, E. E. 1989. BLADES: an artificial intelligence approach to analog circuit design. *IEEE T Comput Aid D*. 8, 6 (Jun. 1989), 680-692. DOI= http://dx.doi.org/10.1109/43.31523

[11] Gielen, G., Wambacq, P. and Sansen, W. M. 1994. Symbolic analysis methods and applications for analog circuits: a tutorial overview. *P IEEE*. 82, 2 (Feb. 1994), 287-304. DOI= http://dx.doi.org/10.1109/5.265355

[12] Gielen, G. G. E. and Rutenbar, R. A. 2000. Computer-aided design of analog and mixed-signal integrated circuits. *P IEEE*. 88, 12 (Dec. 2000), 1825-1852. DOI= http://dx.doi.org/10.1109/5.899053

[13] Hershenson, M. d. M., Boyd, S. P. and Lee, T. H. 1998. GPCAD: a tool for CMOS op-amp synthesis. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design* (San Jose, CA, USA, November 8 - 12, 1998), 296-303. DOI= http://dx.doi.org/10.1109/ICCAD.1998.144281

[14] Hongying, Y. and Jingsong, H. 2010. Evolutionary design of operational amplifier using variable-length differential evolution algorithm. In *Proceedings of the International Conference on Computer Application and System Modeling* (Taiyuan, China, October 22 - 24 2010), 610-614. DOI= http://dx.doi.org/10.1109/ICCASM.2010.5620307

[15] ITRS 2010. International Technology Roadmap for Semiconductors - 2010 Update. ITRS. http://www.itrs.net. Accessed December 10 2011

[16] Koza, J. R., Bennett, F. H., III, Andre, D., Keane, M. A. and Dunlap, F. 1997. Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE T Evolut Comput*. 1, 2 (Jul. 1997), 109-128. DOI= http://dx.doi.org/10.1109/4235.687879

[17] Kruiskamp, W. and Leenaerts, D. 1995. DARWIN: CMOS opamp synthesis by means of a genetic algorithm. In *Proceedings of the ACM/IEEE Design Automation Conference* (San Francisco, California, USA, June 12 - 16, 1995), 433-438. DOI= http://dx.doi.org/10.1109/DAC.1995.249986

[18] Kuo-Hsuan, M., Po-Cheng, P. and Hung-Ming, C. 2011. Integrated hierarchical synthesis of analog/RF circuits with accurate performance mapping. In *Proceedings of the International Symposium on Quality Electronic Design* (Santa Clara, California, USA, March 14 - 16, 2011), 1-8. DOI= http://dx.doi.org/10.1109/ISQED.2011.5770817

[19] Lohn, J. D. and Colombano, S. P. 1999. A circuit representation technique for automated circuit design. *IEEE T Evolut Comput*. 3, 3 (Sep. 1999), 205-219. DOI= http://dx.doi.org/10.1109/4235.788491

[20] Lourenço, N., Vianello, M., Guilherme, J. and Horta, N. 2006. LAYGEN - Automatic Layout Generation of Analog ICs from Hierarchical Template Descriptions. In *Proceedings of the Conference on Ph.D. Research in Microelectronics and Electronics* (Otranto (Lecce), Italy, June 12 - 15, 2006), 213-216. DOI= http://dx.doi.org/10.1109/RME.2006.1689934

[21] Martens, E. and Gielen, G. 2008. Classification of analog synthesis tools based on their architecture selection mechanisms. *Integr VLSI J*. 41, 2 (Feb. 2008), 238-252. DOI= http://dx.doi.org/10.1016/j.vlsi.2007.06.001

[22] Martins, R., Lourenco, N. and Horta, N. 2012. LAYGEN II: Automatic Analog ICs Layout Generator based on a Template Approach. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Philadelphia, USA, July 7 - 11, 2012),

[23] Maulik, P. C., Carley, L. R. and Rutenbar, R. A. 1995. Integer programming based topology selection of cell-level analog circuits. *IEEE T Comput Aid D*. 14, 4 (Apr. 1995), 401-412. DOI= http://dx.doi.org/10.1109/43.372366

[24] McClean, B. 2011. IC Market to Top $300 Billion for First Time in 2013. http://www.icinsights.com/. Accessed December 13 2011

[25] McConaghy, T., Palmers, P., Steyaert, M. and Gielen, G. G. E. 2011. Trustworthy Genetic Programming-Based Synthesis of Analog Circuit Topologies Using Hierarchical Domain-Specific Building Blocks. *IEEE T Evolut Comput*. 15, 4 (Aug. 2011), 557-570. DOI= http://dx.doi.org/10.1109/TEVC.2010.2093581

[26] Ning, Z. Q., Mouthaan, T. and Wallinga, H. 1991. SEAS: a simulated evolution approach for analog circuit synthesis. In *Proceedings of the IEEE Custom Integrated Circuits Conference* (San Diego, California, USA, May 12 - 15, 1991), 5.2-1-4. DOI= http://dx.doi.org/10.1109/CICC.1991.164025

[27] Phelps, R., Krasnicki, M., Rutenbar, R. A., Carley, L. R. and Hellums, J. R. 2000. Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE T Comput Aid D*. 19, 6 (Jun. 2000), 703-717. DOI= http://dx.doi.org/10.1109/43.848091

[28] Roca, E., Velasco-Jimenez, M., Castro-Lopez, R. and Fernandez, F. V. 2010. Context-independent performance modeling of operational amplifiers using Pareto fronts. In *Proceedings of the International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design* (Gammarth, Tunisia, October 4 - 6, 2010), 1-4.

[29] Shoou-Jinn, C., Hao-Sheng, H. and Yan-Kuin, S. 2006. Automated passive filter synthesis using a novel tree representation and genetic programming. *IEEE T Evolut Comput*. 10, 1 (Feb. 2006), 93-100. DOI= http://dx.doi.org/10.1109/TEVC.2005.861415

[30] Sripramong, T. and Toumazou, C. 2002. The invention of CMOS amplifiers using genetic programming and current-flow analysis. *IEEE T Comput Aid D*. 21, 11 (Nov. 2002), 1237-1252. DOI= http://dx.doi.org/10.1109/TCAD.2002.804109