# Creating Flexible Analogue IP Blocks

R. Castro-López, F. V. Fernández, M. Delgado-Restituto, F. Medeiro and A. Rodríguez-Vázquez
*Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica*
*Edif. CICA, Avda. Reina Mercedes s/n, E-41012 Sevilla, SPAIN*
*Phone: +34 955 056 666 FAX: +34 955 056 686*
*E-mail: {castro,pacov,mandel,medeiro,angel}@imse.cnm.es*

## Abstract

*This paper introduces a complete methodology for retargeting of analogue blocks to different sets of specifications. By careful integration of the size tuning of devices and layout generation tasks, fully functional designs are generated in a few minutes of CPU time.*

## 1. Introduction

As chip complexity increases and compressed product development cycles relentlessly scale time-to-market pressures, designers must accomplish more ambitious objectives in less time. For an increasing number of designers, the secret to quickly building highly integrated systems on a chip in a shrinking development cycle lies in the extensive reuse of intellectual property (IP) modules. While a lot of progress has been made in the digital arena in recent years [1], the specific characteristics of analogue design makes the development of flexible analogue IP modules a much more difficult task.

To contribute to the solution of this problem this paper proposes a retargeting for reusability methodology for analogue blocks able to provide working designs for each new set of specifications. The objective of this methodology is not to get the optimum design but to get a design which accomplishes the required specifications in the shortest time.

The methodology is discussed in Section 2. Sections 3 and 4 are devoted to its two main components: the layout generator and the size tuning tool. Finally, Section 5 gives a practical example of retargeting of a fully-differential operational amplifier to a different set of specs.

## 2. Retargeting methodology

The proposed retargeting methodology relies on the previous existence of a block netlist, layout templates for the block at hand, and, optionally, some tuning strategies in the form of design constraints for such block. Retargeting of a given block for a new set of specifications is performed using the iterative mechanism illustrated in Fig.1.

Given a new set of specifications, device sizes are tuned to achieve such specifications. Size tuning is based on an appropriate combination of design rules and constraints, optimization-based sizing using electrical simulation, and constraints from the existing layout templates.

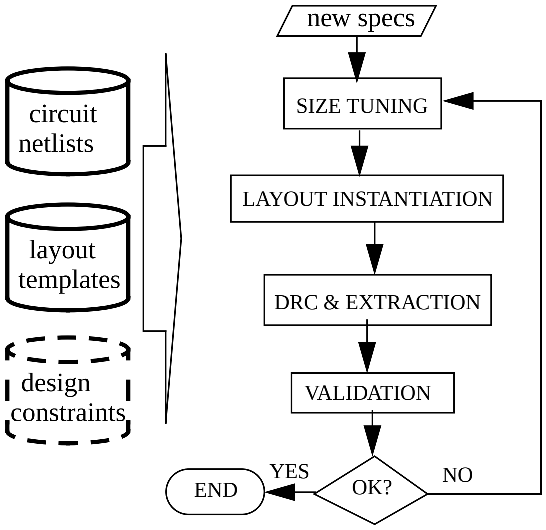

Figure 1. Retargeting cycle.

Resulting device sizes are instantiated on the layout template. Although the layout should be correct by construction, design rules are checked. Layout parasitics are extracted and the circuit performances validated through electrical simulation. In case, some specs are not met, validation information is fed back to the size tuning engine to perform a new iteration.

Usually, a single iteration of the retargeting methodology is sufficient because: (a) the use of a simulation-based approach in the size tuning process guarantees accuracy of the predicted performances; and, (b) the inclusion of layout constraints and parasitic estimations in the tuning procedure since the very first iteration minimizes performance deviations on the extracted layouts.

New methodologies are easier to introduce in existing design flows and designers´ acceptance is improved if they are already familiar with significant parts of it. For this reason, a basic philosophy behind the implementation of the proposed methodology has been to rely on existing commercial tools as much as possible. In particular, many tools of the Cadence Design Framework II environment have been used: Virtuoso Parameterized Cells, SKILL language, design rule checker, layout versus schematic tools, electrical simulator. The only methodology step for which a new tool was required was for performing tuning of device sizes.

## 3. Layout generation

Basically two kinds of approaches have been reported to automatically generate layouts of analogue blocks:

- **Approaches based on capturing the knowledge of expert designers in the form of templates or procedures [2],[3]**. Obviously, they incorporate designer´s expertise on the layout of a given circuit and the instantiation for some specific sizes is very fast. Their main drawbacks are their smaller flexibility and the relatively high cost of the template/procedure generation for each block.
- **Approaches based on formulating the layout generation as an optimization problem [4]-[6]**. In principle, they are fully general. Their main drawbacks are their considerably higher computational cost and their inability to include designers´ expertise. Consequently, they are hardly accepted and none has reached a commercial status.

Taking into account that our objective is to build circuit-specific layout generators and that reuse of designers´ expertise is a major concern we have opted for a parameterized template approach. To palliate the flexibility problem a complete hierarchical parameterization has been pursued together with a strong coupling with the tuning procedures. The template development cost has been reduced by a strong hierarchical decomposition that allows the reuse of subcell layout templates in larger cell templates and by using appropriate structures.

The parameterized layout templates have been built using the Virtuoso Parameterized Cells and the SKILL language [7]. The use of a common commercial framework improves the designers´ acceptance. In addition, although the layout instantiation process is fully automatic, the layout is generated within the commercial tool and, therefore, the designer may perform any modification he/she considers convenient.

When parameterizing complex layout cells, factors such as regularity, density and symmetries must be kept during the retargeting process. This has been achieved by relying on a deep hierarchical decomposition and a careful cell planning. Parameterized layout templates are first built for single devices and small numbers of them (i.e. a set of matching transistors). These basic structures are used to build more complex parameterized subcells, proceeding up the hierarchy until the layout template for the objective block is obtained. During this constructive process much attention is paid to the complete parameterization of cells, relative positions and interconnections, so that, wide changes in device sizes can easily be accomodated. Parameterization of the interconnections does not only consider the design rules but also the current densities that must be carried.

There are layout structures for transistors, capacitors and resistors (and groups of them) more appropriate for the parameterization process. For instance, for a group of transistors with common sources and/or drains and/or gates (i.e. current mirrors, differential pairs) the structure in Fig. 2 [8] is a convenient one. The parameterization is eased because metal lines corresponding to drains, sources and gates are available at both sides of the guard ring, thus, making easier the interconnection of this group of transistors with others and its parameterization. In addition, the structure is a unidirectional common-centroid arrangement, thus, improving transistor matching.

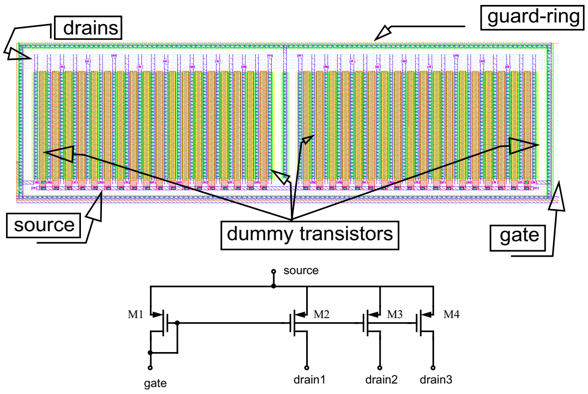

Figure 2. Common-centroid structure for a current mirror.

Two dimensional common-centroid arrangements have also been used for transistors, resistors and capacitors to improve matching considerations.

The construction of a fully-parameterized layout template able to accommodate very different device sizes requires much more effort than the creation of a full-custom layout for a sized circuit, typically about a factor of five times more expensive. The instantiation of the layout template for each new set of device sizes takes less than one second of CPU time and requires no user interaction. Therefore, the additional development cost of the layout template is largely compensated through its repetitive use.

## 4. Size tuning

Design parameters must be appropriately tuned to meet the new design specifications: *restrictions* on the performance of a circuit, and/or *design objectives*. Restrictions are those specifications that include inequalities (i.e. DC-gain > 70dB), and objectives those whose intention is to maximize or to minimize some figure (i.e. minimum power consumption). We will denote acceptability regions those within the multidimensional design space where all design restrictions are met.

Basically two kinds of approaches have been formulated to the sizing problem: knowledge-based and optimization-based. Knowledge-based approaches capture designers´ expertise in the form of some kind of design plans [9],[10]. Knowledge addition is a costly effort, necessary for each particular circuit and not always reusable.

Optimization-based approaches have become much more popular [11]-[15]. The optimization process is an iterative procedure, design parameters being updated at each iteration until an equilibrium point is reached. The degree of compliance of restrictions and design objectives at each iteration is quantified through a cost function.

Within optimization-based systems we can distinguish between equation-based approaches [11],[12] and simulation-based ones [13]-[15]. The former ones use equations to predict circuit performances. They are very efficient to evaluate but they are usually closed systems: an important effort is needed to generate such equations for each new circuit. Moreover, they are usually approximated equations, therefore, yielding suboptimal solutions. Simulation-based approaches are intrinsically open and the predicted performances have the accuracy of the electrical simulator used, at the price of a higher computational cost.

There are basically two optimization alternatives:

- **Deterministic techniques**. An important disadvantage is that only changes of design parameters that decrease the cost function are permitted – the optimization process is quickly trapped in a local minimum of the cost function, so the utility of these techniques concentrates on the fine tuning of suboptimal sizings.
- **Statistical techniques.** The main advantage of the statistical techniques is the capability to escape from local minima, thanks to a nonzero probability of accepting movements that increase the cost function. The price to pay is a larger computational cost.

We have opted for an optimization-based system based on simulation [13]. The implemented approach is a two-step one: in the first step statistical optimization techniques are applied while deterministic ones are applied in the second. A proper formulation of the cost function, and the adjustment of the movement generator to the nature of the analog synthesis drastically decreases the computational cost. To enable the addition of designers´ expertise on tuning procedures for a specific block a mechanism has been developed which allows the user to establish design constraints, which are considered in the optimization process. In general, the design space for a given circuit contains several acceptability regions and each point in these regions is associated to different values of the design objectives. The instantiation of such solutions of the design space in the layout template yields layout instances of different quality. As layout quality is also a major concern, an evaluation is performed at each iteration and contributes to the control of the evolution of the optimization process. This evaluation of layout quality includes from simple size deviations with respect to an ideal instantiation of the layout template to more complex relationships based on the study of the particular layout template. Parasitics estimations are also included in this evaluation.

## 5. A retargeting example

The methodology presented in this paper will be illustrated via the retargeting of the fully-differential Miller-compensated two-stage amplifier in Fig. 3 for a different set of specifications. The circuit was originally designed for the set of specifications in Table 1. The design of the original circuit served to develop some circuit-specific design constraints (i.e., relationships of design parameters to ensure enough current in the current sources of the folded-cascode amplifier in the first stage to drive the transistors under maximum slew-rate conditions).

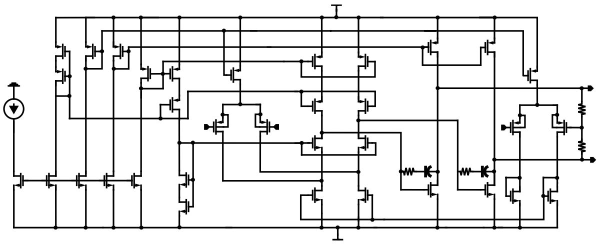

Figure 3. Fully-differential operational amplifier.

Table 1. Specs of original design.

| | Specs | Simulated | Units |
|---|---|---|---|
| $A_0$ | >80 | 87.8 | dB |
| *GBW* | >35 | 35.1 | MHz |
| *PM* | >45 | 47.2 | ° |
| *OS* | >5 | 5.3 | V |
| power | minimize | 1.73e-3 | W |

The sized circuit was also used as a reference to build a layout template. The cell layout was planned to allow a maximum flexibility to the component devices to accommodate the needed new sizes in a cell retargeting process.

Then, the circuit was retargeted to the set of specifications in the first columns in Table 2 in a 0.35µm technology. To illustrate the importance of including the evaluation of the layout quality during the size tuning process, two retargeting experiences were carried out. In the first one, the only objective was to achieve the new set of specifications. The instantiation of the resulting device sizes in the layout template yielded the layout in Fig. 4. The extracted layout met the specifications but, as can be observed, the layout density has largely been deteriorated.

In the second retargeting experience the impact on the layout quality was included in the cost function guiding the optimization process. As can be observed in the instantiation of the sizes in the layout template in Fig. 5, the layout quality and the area efficiency are much better (both layouts in Fig. 4 and Fig. 5 have been captured with the same resolution, therefore, the figures show the actual relative dimensions of both cells). The simulation results of the extracted layout are shown in the third column in Table 2. Notice that the specifications are also met in this new retargeted design.

Only one iteration of the retargeting methodology was needed. The total CPU time of the retargeting process was 5 minutes on a SUN Ultra 10 at 333 MHz. Four of them were spent on the size tuning task while the rest was spent on layout instantiation, DRC, extraction, LVS and final simulation.

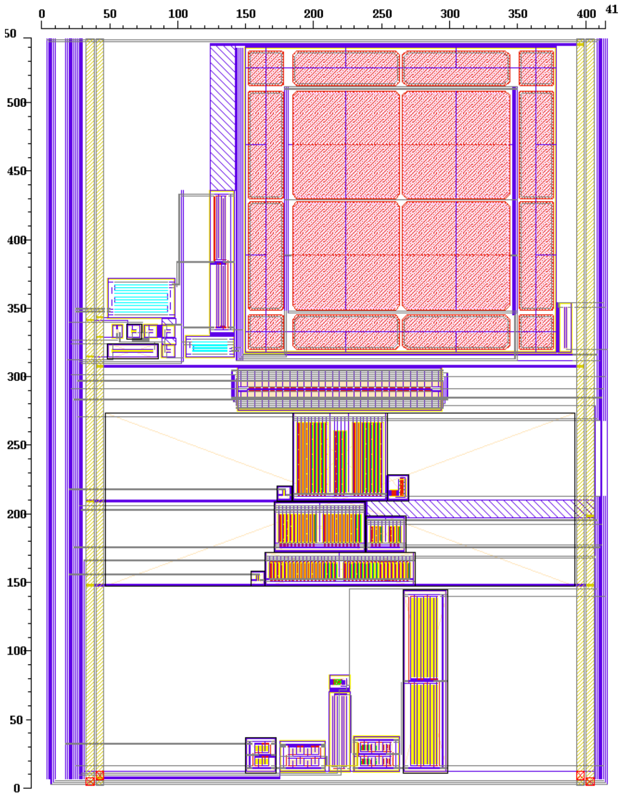

Figure 4. Instantiation of the retargeted design without evaluation of impact on layout quality.

Table 2. Specs and results of retargeted design.

| | Specs | Simulated | Units |
|---|---|---|---|
| $A_0$ | >85 | 87.5 | dB |
| *GBW* | >100 | 100.2 | MHz |
| *PM* | >50 | 50.4 | o |
| *OS* | >5 | 5.01 | V |
| power | minimize | 1.93e-3 | W |

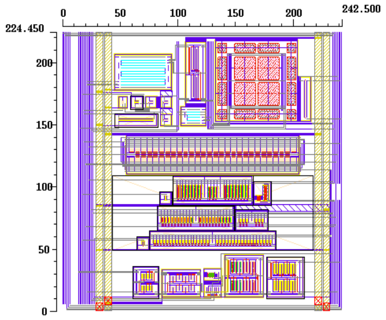

Figure 5. Instantiation of the retargeted design with evaluation of impact on layout quality.

## 6. Conclusions

Lack of flexibility of analogue IP blocks limit their applicability in application scenarios demanding high circuit performances. Through the introduction of a retargeting methodology for transistor level circuits, this paper has tried to contribute in making flexible analogue IP blocks a reality.

## 7. Acknowledgments

This work has been partially supported by the EU ESPRIT Project 29648 RAPID.

## 8. References

[1] R. Seepold, Ed., “Virtual Socket Interface Alliance,” *Proc. Design, Automation and Test in Europe*, p. 182, 1999.

[2] J.D. Conway and G.G. Schrooten, “An Automatic Layout Generator for Analog Circuits,” *Proc. European Design Automation Conf.*, pp. 513-519, 1992.

[3] H. Onodera, H. Kanbara and K. Tamaru, “Operational-Amplifier Compilation with Performance Optimization,” *IEEE J. Solid-State Circuits*, Vol. 25, pp. 466-473, April 1990.

[4] K. Lampaert, G. Gielen and W. Sansen, “A Performance-Driven Placement Tool for Analog Integrated Circuits,” *IEEE J. Solid-State Circuits*, Vol. 30, No. 7, pp. 773-780, July 1995.

[5] J.M. Cohn, R.A. Rutenbar and L.R. Carley, "KOAN/ANAGRAM II: New Tools for Device-Level Analog Placement and Routing," *IEEE J. Solid-State Circuits*, Vol. 26, No. 3, pp. 330-342, March 1991.

[6] E. Malavasi et al., “Automation of IC Layout with Analog Constraints,” *IEEE Trans. Computer-Aided Design*, Vol. 15, No. 8, pp. 923-942, Aug. 1996.

[7] “*Virtuoso Parameterized Cells Reference Manual*” and “*Custom Layout SKILL Functions Reference Manual*”, Product Version 4.4.1, Cadence Design Systems, Feb. 1997.

[8] R. Naiknaware and T.S. Fiez, “Automated Hierarchical CMOS Analog Circuit Stack Generation with Intramodule Connectivity and Matching Considerations,” *IEEE J. Solid-State Circuits*, Vol. 34, No. 3, pp. 304-317, March 1999.

[9] M. Degrauwe et al., "IDAC: An Interactive Design Tool for Analog CMOS Circuits," *IEEE J. of Solid-State Circuits*, Vol. 22, No. 6, pp. 1106-1116, Dec. 1987.

[10] R. Harjani, R.A. Rutenbar and L.R. Carley, "OASYS: A Framework for Analog Circuit Synthesis," *IEEE Trans. on CAD*, Vol. 8, No. 12, pp. 1247-1265, Dec. 1989.

[11] H.Y. Koh, C.H. Sequin and P.R. Gray, "OPASYN: A Compiler for MOS Operational Amplifiers," *IEEE Trans. on CAD*, Vol. 9, No.2, Feb. 1990.

[12] G. Gielen, H. Walsharts and W. Sansen, "Analog Circuit Design Optimization based on Symbolic Simulation and Simulated Annealing," *IEEE J. Solid-State Circuits*, Vol. 25, No. 3, June 1990.

[13] F. Medeiro et al., “Global Design of Analog Cells Using Statistical Optimization Techniques,” *Analog Integrated Circuits and Signal Processing*, Vol. 6, No. 3, pp. 179-195, Nov. 1994.

[14] E. Ochotta, R. Rutenbar and L.R. Carley, “Synthesis of High-Performance Analog Circuits in ASTRX/OBLX”, *IEEE Trans. on CAD*, Vol. 15, pp. 273-294, March 1996.

[15] R. Phelps et al., “Anaconda: Simulation-Based Synthesis of Analog Circuits Via Stochastic Pattern Search,” *IEEE Trans. on CAD*, Vol. 19, No. 6, pp. 703-717, June 2000.