A Constraint-Driven Placement Methodology for Analog Integrated Circuits

Edoardo Charbon, Enrico Malavasi, Umakanta Choudhury, Andrea Casotto,
and Alberto Sangiovanni-Vincentelli
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720

A new constraint-driven methodology for the placement of analog IC's is described. Electrical performance specifications are automatically translated into constraints on the layout parasitics. These constraints and the sensitivity information of the circuit are then used to control a Simulated Annealing-based placement algorithm. At each step of the annealing a fast check on performance degradations is performed to guarantee that the tool has the necessary robustness.

1 Introduction

Layout design automation has generated considerable interest in recent years. The availability of technologies allowing on-chip integration of mixed analog/digital circuits and the increasing importance of ASIC's has determined the emergence of tools targeted toward analog synthesis. Digitally targeted tools are often inadequate to handle the critical and specific requirements of analog layout. The performances of analog circuits are much more sensitive to the details of physical implementation than the digital ones. Custom design requires great flexibility, and layout synthesis is often a multiple-objective optimization problem, where, along with area, wiring length and delay, two groups of relevant issues must be taken into account:

1. Topological constraints, i.e. symmetries and matching.
2. Parasitics associated with devices and interconnections.

Several approaches to automate analog layout have been reported. In most of them typical methodologies derived from the digital world have been used. In [1], coupling between mutually sensitive nets was reduced by means of a standard-cell placement strategy. In [2] a slicing structure min-cut algorithm was used to reduce the CPU time needed for the placement phase of analog IC's. Typically these approaches have not achieved the same layout density and flexibility obtained by expert designers. Others [3] suggested a knowledge-based approach, or required the user to provide basic hints on the layout structure [4]. In [5], very effective analog-specific placement and routing tools accounting for topological constraints have been provided for full-custom layout. A routing methodology for the minimization of cross-coupling has been proposed. However, no method is given to define the actual bounds for the parasitics introduced by the layout.

A fundamental limitation of all these approaches is that physical design is performed with no explicit reference to the performance specifications of the circuit. As a result a large number of time-consuming layout-extraction-simulation iterations may be necessary to meet a set of performance specifications. Only recently performance-driven approaches to the layout of analog circuits have been presented. In [6] algorithms were proposed for generating a set of low-level parasitic constraints from high-level performance constraints. Area routing and channel routing tools based on this methodology have been described in [7] and [8] respectively. More recently the methodology has been generalized to the design of analog IC's in [9].

In this paper we present a placement tool based on the general performance-driven methodology. Automatically generated constraints are used to control a Simulated Annealing placement algorithm, thus trying to insure that the resulting configuration meets all the high-level constraints on performances. Simulated Annealing has been chosen because of the many constraints and cost functions to be included in analog layout that make the use of other placement algorithms very difficult if not impossible. In addition, we have experience in deriving and implementing Simulated Annealing versions for a variety of problems [10, 11, 12, 13]. The algorithm has been used before for analog layout [5], however our approach is significantly different in the way the constraints and cost functions are derived from high-level specifications and in the *accurate* estimation of layout dependent performances.

Our placement approach is illustrated in Figure 1. The constraint generator detects critical nodes and generates a set of bounding constraints for the parasitics associated with them. The cost function minimized by the annealing algorithm accounts for parasitic control, symmetries, matching and well separation. Device abutment, often used by expert designers as a way of maximizing the performances of analog circuits, is performed automatically during the annealing. The objective is not only area and wiring minimization, but especially the elimination of parasitic constraint violations. Multiple symmetry axes can be managed with the algorithm of Virtual Symmetry Axes [14], that dynamically defines axis positions. Thus a more compact layout, less routing congestion and a better matching between the modules is achieved. Finally a verification phase, based on accurate parasitics extraction, checks that all performances be within the required specifications. An example of a layout generated using our methodology is shown in section 4.

2 Simulated Annealing for Analog Placement

Simulated Annealing [15, 16], is a nondeterministic iterative optimization algorithm. With a sufficiently large number of iterations it converges to a configuration with a low cost function value by attempting random configuration perturbations, called *moves*. A move is accepted with probability 1 if it leads to a better configuration, i.e. a state with lower cost. Otherwise, the move is accepted with probability given by

$$\textbf{Prob} \quad = e^{-\frac{\triangle F}{T}}$$

where $\triangle F$ is the cost increase due to the move. T is a parameter called *temperature*, decreasing from a starting value T_0 to a final T_f with an assigned monotonic *cooling schedule*.

A version of Simulated Annealing for the placement of digital macrocells has been implemented in a tool called PUPPY [10]. Its cost function has been modified to include parameters corresponding to analog requirements on parasitic control as well as topology constraints, such as symmetries and device matching. What follows is a description of the modified cost function.

Let S be the space of all placement configurations and let s be an element of such space. The cost function is the weighted sum of seven components:

$$\begin{aligned} v(s) \quad = \quad & \alpha_{wl} f_{wl}(s) + \alpha_a f_a(s) + \alpha_{ov} f_{ov}(s) + \alpha_{sy} f_{sy}(s) \\ & + \alpha_{ma} f_{ma}(s) + \alpha_{we} f_{we}(s) + \alpha_{co} f_{co}(s) \end{aligned} \tag{1}$$

where:

- $f_{wl}(s)$ is the wire length estimated by means of spanning trees.
- $f_a(s)$ is the total area of the circuit. Space for routing is estimated with the *halo* mechanism [12]
- $f_{ov}(s)$ is the total overlapping area between cells.
- $f_{sy}(s)$ is the *distance* between placement s and a symmetric configuration in the metric of the state space, given by:

$$f_{sy}(s) = \sum_{symmetric\ modules} (d_i^{(s)} + W_\rho \rho_i) \tag{2}$$

  Item $d_i^{(s)}$ is the translation needed to bring the i-th cell to a symmetric position. Parameter ρ_i has value 1 if mirroring and/or rotation is needed to enforce symmetry, and 0 otherwise. Weight W_ρ is a parameter whose value is $\gg 1$. In our tool $W_\rho = 10$.
- $f_{ma}(s)$ is a measure of the mismatch between circuit devices:

$$f_{ma}(s) = \sum_{matched\ devices} (d_i^{(m)} + W_\rho \rho_i) \tag{3}$$

  Item $d_i^{(m)}$ is the translation needed to bring the i-th device inside an area of adequate matching characteristics with the other matched devices. This area can be user-specified or automatically computed by our performance-driven approach, as explained in section 3. Parameters ρ_i and W_ρ have the same meaning as in (2).
- $f_{we}(s)$ is a measure of the discontinuity of well regions. This parameter is used only in device-level placement.
- $f_{co}(s)$ is a function accounting for performance constraint violations. Its computation is a key issue in our performance-driven approach and will be detailed in section 3.
- $\alpha_{wl}, \alpha_a, \alpha_{ov}, \alpha_{we}, \alpha_{sy}, \alpha_{ma}$ and α_{co} are non-negative weights. Their initial default values are adjusted dynamically during the algorithm using heuristics aimed at the total elimination of constraint violations at the end of the process.

3 The Enforcement of Performance Constraints

High-level performance constraints are expressed in terms of the maximum degradation allowed on each performance with respect to its nominal value. Degradations are due to parasitics, i.e. stray resistances, capacitances toward substrate and cross-coupling capacitances. Absolute values as well as mismatches of these parameters may have different relevance depending on the sensitivity of performances with respect to them. The maximum values parasitics can assume in a circuit so as not to violate the constraints on the specifications depend on sensitivities as well as on the tightness of constraints.

During placement, capacitances associated with N/P diffusions can be computed accurately based on the structure of the layout modules. These capacitances can be reduced substantially by device abutment. Net capacitances, on the contrary, are more difficult to estimate, since they depend on routing details. However, for a given placement conservative estimates of minimum and maximum values of capacitances can be made on the ground of net length estimates and the available routing layers. Hence, minimum and maximum values can be derived for performance degradations during each annealing iteration. If the maximum degradation for a given performance is below the bound given by the specifications, then no cost function penalty is imposed. If the bound is between the minimum and maximum degradations, a contribution proportional to the constraint violation, i.e. the difference between the maximum degradation and the bound, is added to the cost function. In this case routing has chances to yield a layout meeting the constraints. When the minimum possible degradation exceeds the bound, constraints cannot be met during routing and the placement is not acceptable.

Penalization of the configurations where performances don't meet their constraints is carried out by means of parameter f_{co} introduced in (1). Its expression is computed as follows. Let N_c and N_p be respectively the number of performance constraints and the number of critical parasitics. Let $S_{i,j}$ be the sensitivity of performance K_i with respect to parasitic p_j. Performance degradations are estimated based on linearized expressions using sensitivities. This is an acceptable approximation as long as we assume that performance degradations are small compared to their nominal values. In the general problem formulation, performance constraints can be modeled by the following inequalities:

$$\begin{aligned} \sum_{j=1}^{N_p} \left(S_{i,j}^{+} p_j \right) &\leq \overline{\triangle K_i}^{+} \\ \sum_{j=1}^{N_p} \left(S_{i,j}^{-} p_j \right) &\leq \overline{\triangle K_i}^{-} \end{aligned}$$

where $S_{i,j}^{+}$ is the worst case positive value of $S_{i,j}$, $S_{i,j}^{-}$ is absolute value of the worst case negative value of $S_{i,j}$ [17]. $\overline{\triangle K_i}^{+}$ and $\overline{\triangle K_i}^{-}$ are the constraints on the degradation of performance K_i in the positive and negative direction respectively. These two constraints can be different and one of them can eventually be infinite. In what follows the '+' and '−' signs have been omitted in the notations of sensitivities and constraints, since expressions are given for either positive or negative directions.

Let $p_j^{(min)}$ and $p_j^{(max)}$ be respectively the estimates of the minimum and maximum values of parasitic p_j. If we consider capacitance toward substrate, estimates are based on a model where the minimum-width unit-length capacitance C_u is the sum of a fringe component C_{fr} and a parallel-plate component proportional to the wire width w:

$$C_u = C_{fr} + C_{pp} w.$$

If we consider cross-coupling capacitances, estimates are based on a model where the cross-coupling capacitance C_c is the sum of the fringe components C_{fr1} and C_{fr2} of the two lines, and the overlapping capacitance between the crossing wires C_{cc}.

$$C_c = C_{fr1}(w_2) + C_{fr2}(w_1) + C_{cc} w_1 w_2.$$

These components can be determined for each routing layer with numerical simulations [18]. The estimation of parasitics also takes into account the junction capacitances of interconnected transistors. This component is relatively small if compared with the capacitance due to interconnect lines. However in case of short interconnects lines, it becomes dominant, hindering further parasitic reduction by reducing the size of the wire. A drastic reduction of this parasitic component is possible only through device abutment. In our tool device abutment is performed dynamically during the annealing algorithm as in [5]. In our approach however the process is driven by the parasitic constraints, as well as by area and wiring considerations. Instead of randomly choosing the devices to abut, the algorithm tries first to remove the nets whose parasitics induce a violation on the performance constraints. The move is then accepted according to the annealing paradigm. This non-blind abutment strategy is consistent with our general constraint-driven methodology.

The computed values of C_u and C_c are then used for the computation of the performance degradation. Let ρ_c be the ratio between the maximum and minimum value of C_u. Then $\rho_c = \frac{p_j^{(max)}}{p_j^{(min)}}$. Degradation $\triangle K_i$ of performance K_i is bounded as follows:

$$\triangle K_i^{(min)} \leq \triangle K_i \leq \triangle K_i^{(max)}$$

where

$$\triangle K_i^{(min)} = \sum_{j=1}^{N_p} \left(S_{i,j} p_j^{(min)} \right)$$
$$\triangle K_i^{(max)} = \sum_{j=1}^{N_p} \left(S_{i,j} p_j^{(max)} \right)$$

Function f_{co} is computed as the sum of contributions due to the violations of all constraints:

$$f_{co} = \sum_{i=1}^{N_c} C_{K_i}$$

where C_{K_i} is given by the following expression : (see Figure 2) :

$$C_{K_i} = \begin{cases} 0, & \text{if } \triangle K_i^{(max)} \leq \overline{\triangle K_i} \\ \triangle K_i^{(max)} - \overline{\triangle K_i}, & \text{if } \triangle K_i^{(min)} < \overline{\triangle K_i} \leq \triangle K_i^{(max)} \\ (S_r+1)(\triangle K_i^{(max)} + , & \text{if } \overline{\triangle K_i} \leq \triangle K_i^{(min)} \\ -(S_r \rho_c + 1)\overline{\triangle K_i}) & \end{cases}$$

If $S_r \gg 1$ then the values of C_{K_i} for feasible and unfeasible placements differ by at least one order of magnitude. In our tool $S_r = 10$.

A placement phase (see Figure 1) is followed by a verification phase. Its purpose is to check that no violations on performance constraints exist.

4 Results

The tool implementing the performance-driven approach described in this paper has been implemented using the C language in the UNIX environment. It is currently distributed with the OCT-TOOLS (Rel. 5.1) framework of the University of California, Berkeley.

In Figure 3 the schematic of a folded cascode op-amp is shown. Nominal values and constraints on Low Frequency Gain, Phase Margin and Bandwidth are reported in Table 1. The circuit was placed and routed first introducing the topological constraints only (see Figure 4 and 6). Then the performance constraints were enforced (see Figure 5 and 7). Topological constraints were satisfied in both layouts, however, using the estimation criteria outlined in section 3, PUPPY predicted a violation in two of the three performance constraints in the first layout and none in the second. The routing was performed on both layouts using the same constraints-driven methodology as PUPPY [7]. All parasitics were extracted from the routed layouts and a check on the performance degradation confirmed the prediction (see Table 1).

	L.F.G.	P.M.	U.G.F.
Constraints	0.5 dB	0.5°	50 KHz
Imposing Topological Constraints only	0.3 dB	2.0°	207 KHz
Imposing Topological and Performance Constraints	0.3 dB	0.46°	35.0 KHz

Table 1 Degradations of Low Frequency Gain, Phase Margin and Unity Gain Bandwidth with different placement methodologies.

The placement procedures required a CPU time of 471 and 1203 seconds respectively, on a DECsystem 3100. The results are consistent with an analysis of the layouts. The most critical nets, $10, 11, 12, 13, 14, 15$, (see Figure 3) have been carefully minimized in length when the constraints were introduced. Also no crossings were necessary to route these nets. Abutment was used in four cases to remove constraint violations.

5 Conclusions and Future Work

A constraint-driven approach to analog placement respecting high-level performance specifications has been reported. The algorithm, based on Simulated Annealing, takes into account parasitic control, well separation and enforcement of symmetries and device matching. Device abutment is dynamically performed during the annealing and is explicitly driven by the estimated improvement of the performances due to a drastic decrease of the interconnect and diffusion capacitances, and of the stray resistances at the critical nodes.

A fundamental advantage of our methodology is a drastic reduction of the number of expensive layout-extraction-simulation iterations necessary to meet a set of performance specifications. Results were presented for a real-case circuit example, illustrating the effectiveness of our methodology.

Future work includes the refinement of performance estimation techniques presented in this work. Also alternative approaches to device abutment and symmetry enforcement during the annealing will be investigated. In particular, research has to be done to determine the convergence characteristics of the algorithm when the number of degrees of freedom is dynamically changed.

6 Acknowledgements

This research has been partially supported by DARPA and SRC. The first author has been supported by a grant of The Swiss Science Foundation, Bern, Switzerland. The authors would like to thank Gani Jusuf for providing the examples presented in this work and for useful suggestions during the implementation phase.

References

[1] C. D. Kimble, A. E. Dunlop, G. F. Gross, V. L. Hein, M. Y. Luong, K. J. Stern and E. J. Swanson, "Autorouted Analog

VLSI", in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 72–78, 1985.

[2] J. Rijmenants, J. B. Litsios, T. R. Schwarz and M. G. R. Degrauwe, "ILAC: An Automated Layout Tool for Analog CMOS Circuits", *IEEE Journal of Solid State Circuits*, vol. 24, n. 2, pp. 417–425, April 1989.

[3] M. Kayal, S. Piguet, M. Declercq and B. Hochet, "SALIM: A Layout Generator Tool for Analog ICs", in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 751–754, May 1988.

[4] S. W. Mehranfar, "A Technology-Independent Approach to Custom Analog Cell Generation", *IEEE Journal of Solid State Circuits*, vol. 26, n. 3, pp. 386–393, March 1991.

[5] J. M. Cohn, D. J. Garrod, R. A. Rutenbar and L. R. Carley, "KOAN/ANAGRAM II: New Tools for Device-Level Analog Placement and Routing", *IEEE Journal of Solid State Circuits*, vol. 26, n. 3, pp. 330–342, March 1991.

[6] U. Choudhury and A. Sangiovanni-Vincentelli, "Constraint Generation for Routing Analog Circuits", in *Proc. Design Automation Conference*, pp. 561–566, June 1990.

[7] E. Malavasi, U. Choudhury and A. Sangiovanni-Vincentelli, "A Routing Methodology for Analog Integrated Circuits", in *Proc. IEEE ICCAD*, pp. 202–205, November 1990.

[8] U. Choudhury and A. Sangiovanni-Vincentelli, "Constraint-Based Channel Routing for Analog and Mixed-Analog Digital Circuits", in *Proc. IEEE ICCAD*, pp. 198–201, November 1990.

[9] H. Chang, A. Sangiovanni-Vincentelli, F. Balarin, E. Charbon, U. Choudhury, G. Jusuf, E. Liu, E. Malavasi, R. Neff and P. Gray, "A Top-down, Constraint-Driven Design Methodology for Analog Integrated Circuits", in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 841–846, May 1992.

[10] A. Casotto, F. Romeo and A. Sangiovanni-Vincentelli, "A Parallel Simulated Annealing Algorithm for the Placement of Macro-Cells", *IEEE Trans. on CAD*, vol. CAD-6, n. 5, pp. 838–847, September 1987.

[11] D. F. Wong, H. W. Leong and C. L. Liu, "PLA Folding by Simulated Annealing", *IEEE Journal of Solid State Circuits*, vol. sc-22, n. 2, pp. 208–215, April 1987.

[12] C. Sechen and A. Sangiovanni-Vincentelli, "Chip-Planning, Placement and Global Routing of Macro/Custom Cell IC's using Simulated Annealing", in *Proc. Design Automation Conference*, pp. 73–80, June 1988.

[13] F. Romeo and A. Sangiovanni-Vincentelli, "A Theoretical Framework for Simulated Annealing", *Algorithmica*, March 1991.

[14] E. Malavasi, E. Charbon, G. Jusuf, R. Totaro and A. Sangiovanni-Vincentelli, "Virtual Symmetry Axes for the Layout of Analog IC's", in *Proc. 2^{nd} ICVC, Seoul, Korea*, pp. 195–198, October 1991.

[15] E. H. L. Aarts and P. J. M. van Laarhoven, *Simulated Annealing: Theory and Applications*, D. Reidel Publishing, 1987.

[16] D. F. Wong, H. W. Leong and C. L. Liu, *Simulated Annealing for VLSI Design*, Kluwer Academic, Boston, 1988.

[17] U. Choudhury and A. Sangiovanni-Vincentelli, "Use of Performance Sensitivities in Routing of Analog Circuits", in *Proc. IEEE Int. Symposium on Circuits and Systems*, pp. 348–351, May 1990.

[18] U. Choudhury and A. Sangiovanni-Vincentelli, "An Analytical-Model Generator for Interconnect Capacitances", in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 861–864, May 1991.

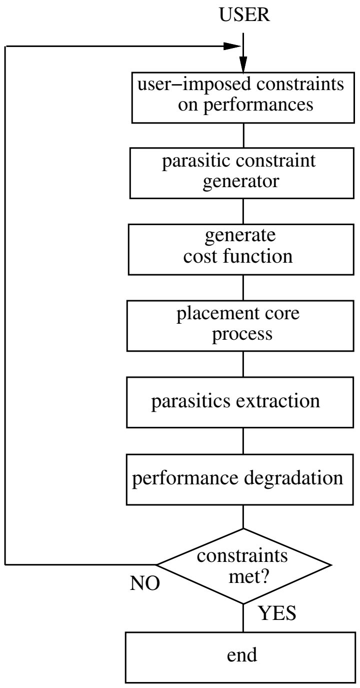

Figure 1: performance-driven placement of analog circuits.

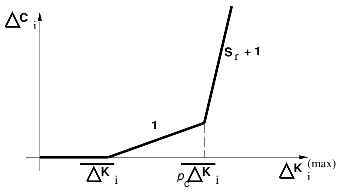

Figure 2: Cost function for component $\triangle K_i$ of f_{co}.

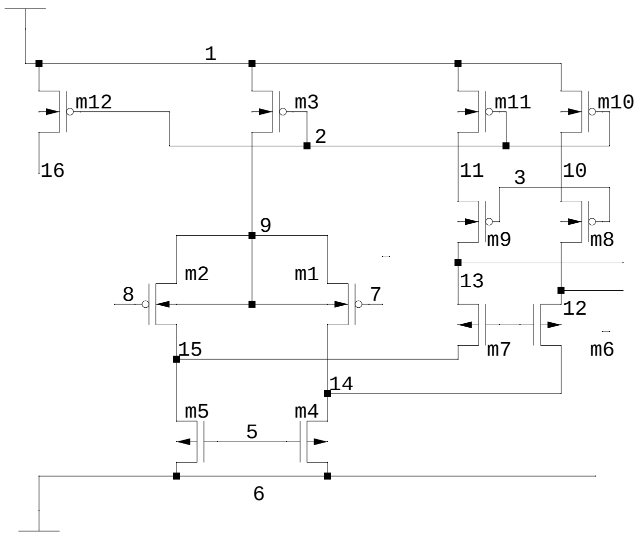

Figure 3: Schematics of the op amp used as example.

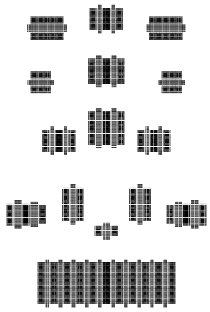

Figure 4: Placement with Topological Constraints only.

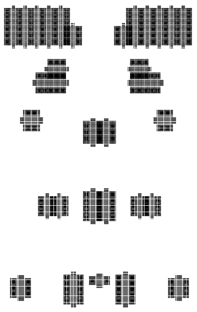

Figure 5: Placement with Topological and Parasitic Constraints.

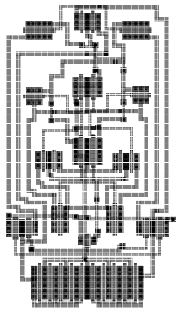

Figure 6: Non-Constrained Placement after Routing.

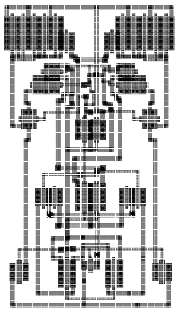

Figure 7: Constrained Placement after Routing.