

# Automation of IC Layout with Analog Constraints

Enrico Malavasi, Edoardo Charbon, Eric Felt and Alberto Sangiovanni-Vincentelli

*Abstract*— A methodology for the automatic synthesis of full-custom IC layout with analog constraints is presented. The methodology guarantees that all performance constraints are met when feasible, or otherwise infeasibility is detected as soon as possible, thus providing a robust and efficient design environment. In the proposed approach, performance specifications are translated into lower level bounds on parasitics or geometric parameters, using sensitivity analysis. Bounds can be used by a set of specialized layout tools performing stack generation, placement, routing and compaction. For each tool, a detailed description is provided of its functionality, of the way constraints are mapped and enforced, and of its impact on the design flow. Examples drawn from industrial applications are reported to illustrate the effectiveness of the approach.

*Keywords*— Layout, Analog Design, Constraint-Driven Layout.

## I. INTRODUCTION

The layout of analog circuits is intrinsically more difficult than the digital one. High performance can be achieved by taking advantage of the physical characteristics of integrated devices and of the correlation between electrical parameters and their variations due to statistical fluctuations of the manufacturing process. Device matchings, parasitics, thermal and substrate effects must all be taken into account. The nominal values of performance functions are subject to degradation due to a large number of parasitics which are generally difficult to estimate accurately before the actual layout is completed. With severe performance degradation, some specifications may not be satisfied, thus jeopardizing the functionality of larger designs of which the circuit is a relevant component.

At the system level, analog silicon compilers have reached satisfactory results with systems characterized by regular hierarchical structures. Examples are programs for the automatic synthesis of opamps and comparators [1], [2], [3], switched-capacitor filters [4], [5], [6] and data converters [7], [8], [9]. Although these generators cover a substantial fraction of the analog circuits needed in most industrial applications, a more general approach, able to cope with arbitrary architecture and full custom layout, is often needed. A variety of approaches inherited from the digital CAD world, with placement based on slicing structures

and channel routing, have been proposed: in macro-cell [10] and standard-cell [11], [12], [13] approaches, capacitive coupling between interconnections is minimized during global routing by allocating sensitive nets to separate channels. In ILAC [14], the layout generator for the analog synthesis system IDAC [15] the layout is based on the generation of specialized pre-defined modules. Knowledge-based approaches, such as the ones presented in SALIM [16], LADIES [17] and BLADES [18], rely heavily on the user's expertise. In [19], a gridless channel-router is described, where great importance is given to parasitic control during routing: capacitance bounds between nets are preserved by setting the minimum separation between horizontal segments in a channel, and by ordering them to avoid crossovers (if possible). In [20], cross-coupling minimization is the routing target, while stray resistances are controlled by means of variable wire-segment widths. In STAT [21] and in KOAN/ANAGRAM [22], placement and routing rely on weighted parasitic minimization and matching constraint enforcement. Area routing and unconstrained placement with abutment capability provide the layout with high flexibility and good area performance. However, no clear strategy is indicated for the definition of parasitic weights, of routing schedule and matching constraints. This information must be supplied by the user, on the ground of his/her experience and knowledge of the circuit behavior. Finally, most of the systems cited above produce non-compacted layouts. Unconstrained compaction can degrade parasitics by modifying the spacings between interconnections and matched devices. So far, few approaches [23] have been reported which face, at least in part, the multi-constrained analog compaction problem.

A set of tools able to guarantee that constraints are met in a reasonable number of applications would have a considerable impact, since analog designers need to trust a tool to meet their specs before using it. Analog CAD tools, like their digital counterparts, must guarantee to meet all specs, or otherwise to detect as soon as possible infeasibility and its causes. Only recently constraint-driven layout generation tools [24], [25], [26], [27] have been proposed, generally based on sensitivity analysis of circuit performance [28], [29]. In this paper, a methodology and the supporting tools [30] for performance-driven layout synthesis are presented. In the methodology, high-level constraints are automatically translated into a set of low-level bounds on the parameters (i.e. parasitics and geometry) that can be effectively controlled during layout synthesis. Design choices are taken trying to detect infeasible configurations as early as possible. After each stage of the design, further elaboration is allowed only if the partial design can meet all

E. Malavasi is currently with Cadence Design Systems, San Jose, CA, and on leave from the *Dipartimento di Eletttronica e Informatica*, University of Padova, Italy.

E. Charbon, E. Felt and A. Sangiovanni-Vincentelli are with the Dept. of Electrical Engineering and Computer Science, University of California, Berkeley.

This research has been supported in part by SRC (grant 91-DC-008), ARPA (grant J-FBI-90-073), FUJITSU, the MICRO Program of the State of California, the Italian National Council of Research, and Asea Brown Boveri, Baden, Switzerland.

performance specifications. While the tools supporting the methodology have been presented [28], [27], [31], [32], [33], [34], so far no paper has been published to present an overall view of the layout methodology. Moreover, to the best of our knowledge this paper is the first comprehensive presentation of a fully integrated performance-driven analog layout system and of the tools supporting it, targeted toward a general set of applications.

The top-down layout design flow is illustrated in Fig.1. At each step, the existence of a feasible configuration is checked and high-level constraints are translated into a set of bounds on low-level parameters. Sensitivity analysis and parasitic estimates are used to determine feasible bounds. Among all the possible sets of bounds, the one maximizing the *flexibility* of the tool to be used is chosen. Flexibility is a function which measures how easily the tool is able to meet the given set of constraints. In feedback paths, infeasible solutions are analyzed to increase the accuracy of the parameter estimates used for bound generation.

Each layout phase is organized as illustrated in Fig.2. The design task is constrained by a set of input specifications, which are either high-level performance specifications or additional design constraints introduced by other layout phases. Constraints are translated into a set of bounds on parasitics by a *constraint generator*, based on estimates of the feasible values of each parasitic. These bounds drive each tool independently. The resulting layout is then analyzed to check whether performance specifications have actually been met. If some constraint has been violated, the values of the extracted parasitics can provide more accurate estimates to the constraint generator. The constraint generator also executes the feasibility check. In fact, low-level bounds must be feasible, i.e. they must lay between the minimum and maximum possible values estimated for the parameters. Such early detection of infeasibility provides an efficient control of design iterations, thus minimizing overall computation time. Feedback control paths provide previous design phases with information on those critical parasitics for which it was not possible to determine feasible bounds with the current configuration.

In this paper, several tools supporting the methodology are described. The basic algorithms employed by the tools are described, while detailed discussion on each tool can be found in the referenced literature. Emphasis is given to techniques and algorithms for the management of analog constraints, and to their coordination in the design flow. The organization of this paper is as follows. Section II presents an overview of sensitivity analysis and on the techniques for constraint translation to low-level bounds. The constraint generator PARCAR is described in Section III. In the following sections, the constraint-driven tools are described. The placement tool PUPPY-A is described in Section IV, the routers ART and ROAD are presented in Section V, and the compactor SPARCS-A is in Section VI. Experimental results on industrial-strength benchmarks are reported in Section VII, followed by conclusions in Section VIII.

## II. SENSITIVITY ANALYSIS AND CONSTRAINT GENERATION

Constraint generation is the translation of high-level performance specifications into bounds on low-level layout parameters, such as parasitics, wire and device spacing, and symmetries.

High-level performance constraints are expressed as maximum allowed degradations from nominal values, due to process variance and to the parasitics introduced in the definition of layout details. Both absolute parasitic values and mismatch play a role in the deviation of performance functions from their nominal behavior. If some regularity assumptions are satisfied, the relative importance of each parameter can be expressed by the *sensitivity* of performance functions with respect to the parameters.

We denote by  $N_p$  the number of layout parameters, by  $\mathbf{p} = [p_1 \dots p_{N_p}]^T$  the array of all such parameters, and by  $\mathbf{p}^{(0)} = [p_1^{(0)} \dots p_{N_p}^{(0)}]^T$  the array of their *nominal* values. Each performance  $K_i$  is a non-linear continuously differentiable function of all parasitics  $K_i = K_i(\mathbf{p})$  and the array of the  $N_k$  performance functions will be indicated as  $\mathbf{K} = \mathbf{K}(\mathbf{p}) = [K_1(\mathbf{p}) \dots K_{N_k}(\mathbf{p})]^T$ . If all parasitics are subject to variations with respect to their nominal values, let  $\Delta\mathbf{K}(\mathbf{p}) = \mathbf{K}(\mathbf{p}) - \mathbf{K}(\mathbf{p}^{(0)})$  be the corresponding degradation of  $\mathbf{K}$  due to such variations.

A generalized expression for the computation of sensitivities from a set of arbitrary performance functions has been derived in [35], [36]. With this formulation, all performance functions can be represented in a compact and rigorous way, as long as they are continuous and sufficiently regular in an interval around their nominal value, The sensitivity of  $K_i$  with respect to  $p_j$  is defined as<sup>1</sup>

$$S_{i,j} = \left. \frac{\partial K_i(\mathbf{p})}{\partial p_j} \right|_{\mathbf{p}^{(0)}}$$

The array of all sensitivities is

$$\mathbf{S} = \begin{bmatrix} S_{1,1} & \dots & S_{1,N_p} \\ \dots & \dots & \dots \\ S_{N_k,1} & \dots & S_{N_k,N_p} \end{bmatrix}.$$

Sensitivities are computed for each performance function, with respect to all the parameters that may be introduced or modified by the layout phase, i.e. parasitics and geometric parameters. The *adjoint technique* of sensitivity analysis [37] has been used in the AC, DC and time [38] domain. Performance degradations are approximated by linearized expressions using sensitivities [39], which is acceptable as long as we assume that degradations are small compared to the nominal values. The array of all degradations of performance functions due to parasitic variations is

$$\Delta\mathbf{K}(\mathbf{p}) = \mathbf{S} [\mathbf{p} - \mathbf{p}^{(0)}] \quad (1)$$

Before the definition of layout details, one cannot take advantage of the possible cancellation effects due to positive

<sup>1</sup>Here and in what follows the non-normalized notation, first used in [28], is used for sensitivities, without loss of generality.

and negative sensitivities for different parasitics. Hence, each performance constraint is modeled only with respect to the parasitics whose sensitivity is either positive or negative, depending on the sign of the constraint itself. In the general problem formulation, performance constraints are modeled by the following inequalities <sup>2</sup>:

$$\Delta \mathbf{K}(\mathbf{p}) - \overline{\Delta \mathbf{K}^+} \leq \mathbf{0} \quad (2)$$

$$\Delta \mathbf{K}(\mathbf{p}) + \overline{\Delta \mathbf{K}^-} \geq \mathbf{0} \quad (3)$$

where  $\overline{\Delta \mathbf{K}^+}$  and  $\overline{\Delta \mathbf{K}^-}$  are the vectors of constraints, in absolute value, on the degradation of performance functions  $\mathbf{K}(\mathbf{p})$  in the positive and negative direction respectively. They can be different and one of them can eventually be infinite. By substituting the linearized expression (1) in inequalities (2) and (3), the general problem can be rewritten as

$$\mathbf{S}^+ [\mathbf{p} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}^+} \leq \mathbf{0} \quad (4)$$

$$\mathbf{S}^- [\mathbf{p} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}^-} \leq \mathbf{0} \quad (5)$$

where  $\mathbf{S}^+$  is the matrix of the worst-case positive sensitivities and  $\mathbf{S}^-$  is the matrix of the absolute values of the worst-case negative sensitivities:

$$\begin{aligned} \mathbf{S}^+ \{i, j\} &= \max(0, S_{i,j}) \\ \mathbf{S}^- \{i, j\} &= \max(0, -S_{i,j}) \end{aligned}$$

In the remainder of this paper the '+' and '-' signs have been omitted in the notations of sensitivities and constraints. Expressions (4) and (5) are given for positive and negative directions, and the general problem formulation becomes

$$\mathbf{S} [\mathbf{p} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}} \leq \mathbf{0}. \quad (6)$$

We want to determine an array of *bounds*  $\mathbf{p}^{(\mathbf{b})} = [p_1^{(\mathbf{b})} \dots p_{N_p}^{(\mathbf{b})}]^T$  for all parasitics, such that inequality (6) holds as long as each parasitic remains below its bound, i.e.

$$\mathbf{S} [\mathbf{p}^{(\mathbf{b})} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}} = \mathbf{0}. \quad (7)$$

All bounds must be *feasible* and *meaningful*, i.e. they must be within the range of values that the parasitics can assume in practice. Let  $p_j^{(\min)}$  and  $p_j^{(\max)}$  be respectively the minimum and maximum possible values which can be assumed by parasitic  $p_j$ , and let  $\mathbf{p}^{(\min)} = [p_1^{(\min)} \dots p_{N_p}^{(\min)}]^T$  and  $\mathbf{p}^{(\max)} = [p_1^{(\max)} \dots p_{N_p}^{(\max)}]^T$ . The array of bounds  $\mathbf{p}^{(\mathbf{b})}$  must satisfy the following inequalities:

$$\begin{cases} \mathbf{p}^{(\mathbf{b})} - \mathbf{p}^{(\min)} \geq \mathbf{0} \\ \mathbf{p}^{(\mathbf{b})} - \mathbf{p}^{(\max)} \leq \mathbf{0} \end{cases} \quad (8)$$

The solution of equation (7), subject to the feasibility constraints (8), is called the *constraint-generation problem*.

The notation adopted in this paper, for frequently used parasitics and performance functions, is reported in Table 1.

<sup>2</sup>The notation  $\mathbf{A} \leq \mathbf{0}$  means that every element of array  $\mathbf{A}$  is a real number not greater than 0. Similarly,  $\mathbf{A} \geq \mathbf{0}$  indicates that every element of  $\mathbf{A}$  is a non-negative real number.

### III. THE CONSTRAINT GENERATOR

In general, an infinite number of solutions exist for the constraint-generation problem. PARCAR [28] is a constraint-generator, namely a tool able to find a solution to the constraint-generation problem under particular assumptions. Among all solutions, PARCAR chooses the one maximizing the layout tool *flexibility*, which is a measure of how easily the tool is able to meet the constraints. To explain this concept, suppose that the bound for a given parasitic  $p_j$  is close to its lower limit  $p_j^{(\min)}$ , and far from its upper limit  $p_j^{(\max)}$ . Then the tool is required to maintain  $p_j$  within a bound which imposes a tight limit to its variation. If, on the contrary, the bound is close to  $p_j^{(\max)}$ , the effort required is lower, and the constraint is easier to meet. Therefore flexibility is defined as

$$F = 1 - \frac{\|\mathbf{p}^{(\max)} - \mathbf{p}^{(\mathbf{b})}\|_2}{\|\mathbf{p}^{(\max)} - \mathbf{p}^{(\min)}\|_2}.$$

A discussion of this definition and of the quadratic norm choice can be found in [28]. In PARCAR, a geometric norm is used, and the constraint-generation problem is solved by minimizing a quadratic function (the geometric norm) subject to linear constraints (7) and (8), using a standard quadratic programming (QP) package.

The quality of the result depends on the estimates of parasitic limits  $\mathbf{p}^{(\min)}$  and  $\mathbf{p}^{(\max)}$ , which become more and more accurate as layout details are defined during the design. The values of  $\mathbf{p}^{(\min)}$  and  $\mathbf{p}^{(\max)}$  are generally not known a priori. However, it is possible to compute suitable estimates, which depend on the layout algorithm used. For example, the minimum value of the cross-coupling capacitance between unrouted nets can be set either to 0, or to the crossover capacitance due to unavoidable crossings. The latter estimate, however, is possible only if the router is able to detect unavoidable net crossings. This is the case for a channel router, where wire paths have been predefined in the global routing phase. With maze routing, on the contrary, the minimum value is always set to 0.

A substantial speed-up of the QP solver is achieved by removing from the problem those parasitics whose cumulative contribution to performance degradation is negligible. A threshold value  $\alpha < 1$  is defined (in PARCAR we set  $\alpha = 0.01$ ). For each performance function  $K_i$ , all parasitics are sorted by increasing value. The first  $n_i$  parasitics in the sorted list such that:

$$\sum_{j=1}^{n_i} S_j^i p_j^{\max} \leq \alpha \overline{\Delta K_i} \quad (9)$$

are considered non-critical with respect to the threshold  $\alpha$ . This procedure detects the parasitics whose cumulative contribution to performance degradation is small. To compensate for this simplification in the constraint-generation problem, equation (7) is modified by replacing  $\Delta \mathbf{K}$  with  $(1 - \alpha)\Delta \mathbf{K}$ . Notice that the sorting order may be different for different performance functions. Let  $P_i$  denote the set

of  $n_i$  critical parasitics sorted according to performance  $K_i$ . When all performance functions are considered simultaneously, the set of non-critical parasitics is

$$P = \bigcap_{i=1}^{N_k} P_i.$$

The set  $P$  of non-critical parasitics determined in this way is eliminated from further analysis. Different sorted lists are maintained for each kind of parasitics (capacitances, resistances and inductances) and elimination is carried out separately. This simplification can be very effective, since in most cases it allows to eliminate a relevant number of negligible parasitics.

### Matching Constraints

Matching constraints are drawn from high-level constraints through sensitivity analysis. Matching of devices or interconnections can be defined as a correlation enforced between their electrical parameters, by means of a proper layout setup minimizing the effect of technological gradients and random mask errors.

Consider two parasitics  $p_1$  and  $p_2$ . Within the limits of linear approximation (1), their contribution to the degradation of performance  $K_i$  is

$$\Delta K_i|_{1,2} = S_{i,1}p_1 + S_{i,2}p_2 = 2S_{i,p}p + \frac{S_{i,\Delta}}{2}\Delta p \quad (10)$$

where

$$p = \frac{p_1 + p_2}{2} \quad S_{i,p} = S_{i,1} + S_{i,2}$$

$$\Delta p = p_1 - p_2 \quad S_{i,\Delta} = \frac{S_{i,1} - S_{i,2}}{2}$$

It is evident that if

$$\left| \frac{S_{i,\Delta}}{S_{i,p}} \right| \gg 1 \quad (11)$$

the contribution of  $p_1$  and  $p_2$  to the degradation of  $K_i$  can be significantly reduced by increasing the correlation between the two parasitics, i.e. by enforcing matching between them. Inequality (11) determines quantitatively the benefit deriving from matching enforcement. For each pair of parasitics, their mismatch and average sensitivities are computed. If relation (11) holds, the mismatch  $\Delta p$  and the average value  $p$  replace  $p_1$  and  $p_2$  in the list of parasitics. In our approach, the magnitude requested to ratio  $\left| \frac{S_{i,\Delta}}{S_{i,p}} \right|$  is user-defined. In our tests, we have obtained good results by requiring the ratio to be at least 10, and this value has been used in all the examples of this paper. If we assume that all parameters in  $\mathbf{p}$  (mismatches as well as parasitics) are independent random variables with zero mean, the variance of the degradation of performance function  $K_i$  with respect to the variances of all mismatches is

$$\sigma^2(\Delta K_i) = \sum_{j \neq i} |S_{i,\Delta_j}|^2 \sigma^2(\delta p_{j,i}). \quad (12)$$

In [40], and more recently in [41], relations have been determined between variances and the relative orientation and

distance between device pairs. This information can be used to translate the maximum allowed performance degradation into constraints on the physical separation and relative orientation between devices. This procedure has been described in detail in [42].

### Symmetry Constraints

A quantitative approach to the determination of all parasitic and device symmetry constraints has been developed and is used to generate automatically symmetry constraints. Symmetry is recognized as a particular case of matching between devices or interconnections belonging to distinct differential signal paths, which become effective when the circuit is operated in differential mode. A graph-based search algorithm, described in detail in [42], has been designed for the automated detection of all critical symmetry constraints. First, a graph is built, with a node for each circuit net, and an edge for each device, to represent the circuit connectivity. Then all virtual grounds are detected by comparing the common- and differential-mode gains of all nets. The search algorithm recognizes all the sub-graphs whose structure has the following characteristics:

1. symmetric topology
2. matching constraints between symmetric graph elements
3. the two halves of the structure are connected with one another by one or more real or virtual ground nets.

Each of these sub-graphs is a differential structure, and the symmetry constraints are all the matching constraints recognized at Step 2.

### Example

As a practical example, consider the clocked comparator COMPL, whose schematic is shown in Fig.3. This comparator has been used as a benchmark in several recent works on analog CAD [32], [43], [44], due to its relevant performance sensitivity to layout details. Consider the following stray resistances (see Table 1 for notation) and the corresponding sensitivities of systematic offset  $V_{off}$  with respect to each of them:

$$\mathbf{p} = \begin{bmatrix} R_{S\_1} \\ R_{S\_2} \\ R_{S\_3} \\ R_{S\_4} \\ R_{S\_6} \\ R_{S\_7} \\ R_{S\_20} \\ R_{S\_21} \\ R_{S\_22} \\ R_{S\_23} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 56.53 \\ -56.53 \\ 0.202 \\ -0.202 \\ 11.83 \\ -11.83 \\ 16.76 \\ -16.76 \\ -16.72 \\ 16.72 \end{bmatrix}^T$$

Offset sensitivities to resistances are expressed in  $\mu V/\Omega$ . They were computed by SPICE-3 [38] with a precision within the third digit. Therefore for each of the pairs  $R_{S\_1,2}, R_{S\_3,4}, R_{S\_6,7}, R_{S\_20,21}, R_{S\_22,23}, R_{S\_21,23}, R_{S\_20,22}, R_{S\_21,22}$  the ratio (11) is  $\left| \frac{S_{i,\Delta}}{S_{i,p}} \right| \geq 10^3$ , i.e. the resistive mismatch is

at least  $10^3$  times more important for offset than the absolute values of these resistances. By simplification (10), offset sensitivities with respect to mismatches become

$$\mathbf{p} = \begin{bmatrix} R_{S_{-1,2}} \\ R_{S_{-20,23}} \\ R_{S_{-21,23}} \\ R_{S_{-20,22}} \\ R_{S_{-21,22}} \\ R_{S_{-6,7}} \\ R_{S_{-3,4}} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 56.53 \\ 16.76 \\ 16.74 \\ 16.74 \\ 16.72 \\ 11.83 \\ 0.201 \end{bmatrix}^T$$

The cumulative effect of all average values on performance degradation is negligible according to (9), and therefore they are all eliminated from  $\mathbf{p}$ .

The symmetry-constraint graph-search algorithm detected the following symmetric net pairs:

$$(52, 53), (15, 16), (10, 11), (13, 14), (55, 56)$$

and the following device pairs:

$$(M_1, M_2), (M_{20}, M_{22}), (M_{21}, M_{23}), \\ (M_{25}, M_{26}), (M_6, M_7), (M_{10}, M_{11}), (M_8, M_9).$$

Performance constraints are enforced on the max switching delay  $\tau_D$  and on systematic offset  $V_{off}$ :

$$\begin{aligned} \tau_D &\leq 7ns \\ |V_{off}| &\leq 1mV \end{aligned} \quad (13)$$

In the first steps of layout, we assume that the nominal value of all parasitics is 0, i.e.  $\mathbf{p}^{(0)} = [0 \dots 0]^T$ . Simulation yields a nominal value of the switching delay  $\tau_D^{(0)} = 4ns$  and null offset. Therefore

$$\mathbf{K} = \begin{bmatrix} \tau_D \\ V_{off} \\ -V_{off} \end{bmatrix} \quad \mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 4.0ns \\ 0.0 \\ 0.0 \end{bmatrix} \quad \overline{\Delta \mathbf{K}} = \begin{bmatrix} 3.0ns \\ 1mV \\ 1mV \end{bmatrix}$$

As expected, sensitivity analysis shows that delay is sensitive to stray capacitances, while resistances and mismatch affect only offset:

$$\mathbf{p} = \begin{bmatrix} C_{15} \\ C_{16} \\ C_{55} \\ C_{56} \\ R_{S_{-1,2}} \\ R_{S_{-20,23}} \\ R_{S_{-21,23}} \\ R_{S_{-20,22}} \\ R_{S_{-21,22}} \\ R_{S_{-6,7}} \\ R_{S_{-3,4}} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 36ps/fF & 0.0 & 0.0 \\ 36ps/fF & 0.0 & 0.0 \\ 47ps/fF & 0.0 & 0.0 \\ 47ps/fF & 0.0 & 0.0 \\ 0.0 & 0.056mV/\Omega & 0.056mV/\Omega \\ 0.0 & 0.016mV/\Omega & 0.016mV/\Omega \\ 0.0 & 0.011mV/\Omega & 0.011mV/\Omega \\ 0.0 & 0.201\mu V/\Omega & 0.201\mu V/\Omega \end{bmatrix}^T$$

Because of symmetries, and since the nominal value of mismatch is 0, offset sensitivities in the positive and negative direction are equal. We use the following conservative minimum and maximum parasitic estimates:

$$\begin{aligned} \min C &= 1fF \\ \max C &= 100fF \\ \min R &= 0 \\ \max R &= 50\Omega \end{aligned}$$

With these estimates, PARCAR computed the following set of parasitic bounds:

$$\mathbf{p}^{(b)} = \begin{bmatrix} 71.96fF \\ 71.96fF \\ 78.52fF \\ 78.52fF \\ 1.0\Omega \\ 7.4\Omega \\ 7.4\Omega \\ 7.4\Omega \\ 7.5\Omega \\ 19.9\Omega \\ 49.5\Omega \end{bmatrix}$$

Here the relation between sensitivity and tightness of bounds is evident. Only a few parameters affect critically the performance of this circuit and therefore need to be bounded tightly. In practice, only the mismatch between the source resistances in the differential pair and the mismatch between the two current mirrors ( $M_{20}, M_{23}$ ) and ( $M_{21}, M_{22}$ ) are responsible for offset.

#### IV. PLACEMENT WITH ANALOG CONSTRAINTS

PUPPY-A [27] is a macro-cell-style placement tool based on Simulated Annealing (S.A.) [45]. In PUPPY-A, the cost function is a weighted sum of non-homogeneous parameters controlling parasitics, symmetries and device matching. Let  $\mathbf{s}$  be a placement configuration, i.e. the set of the positions and rotation angles of all layout modules. The cost function is given by the following expression:

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

where:

- $f_{wl}(\mathbf{s})$  is the sum of wire length estimates over all the modules. Two estimation methods are available, one based on semi-perimeter and the other on pseudo-Steiner tree technique.
- $f_a(\mathbf{s})$  is the total area of the circuit. Space for routing is estimated with the *halo* mechanism described in [46]
- $f_{ov}(\mathbf{s})$  is the total overlapping area between cells.
- $f_{we}(\mathbf{s})$  is a measure of the discontinuity of well regions. This parameter is used only in device-level placement. It is given by the sum of the distances between devices that should lay within the same well or substrate region.
- $f_{sy}(\mathbf{s})$  is a measure of the *distance* between placement  $\mathbf{s}$  and a symmetric configuration, given by the following expression:

$$f_{sy}(\mathbf{s}) = \sum (d(\mathbf{s})_i + \rho_i) \quad (14)$$

where the sum is extended to all symmetric devices. Item  $d(\mathbf{s})_i$  is the translation needed to bring the  $i$ -th cell to a symmetric position. The value of  $\rho_i$  is 0 if mirroring and/or rotation are not needed to enforce symmetry, otherwise it is set to 10.

- $f_{ma}(\mathbf{s})$  is a measure of the mismatch between circuit devices. Its definition is similar to the one of  $f_{sy}(\mathbf{s})$ :

$$f_{ma}(\mathbf{s}) = \sum (d_i^{(m)} + \rho_i)$$

where the sum is extended to all matched devices. 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 as explained in Section III. Parameter  $\rho_i$  and has the same meaning as in (14).

- $f_{co}(\mathbf{s})$  is a penalty function accounting for performance constraint violations. Its computation is key to our performance-driven approach. Estimates  $\mathbf{p}^{(\min)}$  and  $\mathbf{p}^{(\max)}$  of minimum and maximum interconnect capacitances and resistances are obtained on the ground of net length estimates and of the available routing layers. Using the linearized expression (1), performance degradation can be computed at each annealing iteration, and one of the following cases can apply.
  1. If the maximum degradation is within the specifications, that is

$$\mathbf{S} [\mathbf{p}^{(max)} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}} \leq \mathbf{0} \quad (15)$$

no cost function penalty is imposed. In fact, in this case constraints (7) and (8) are met whatever values the parasitics assume.

2. Otherwise,  $f_{co}(\mathbf{s})$ , is a function of the constraint violation  $\Delta \mathbf{K}(\mathbf{p})$ .

In case 2, the penalty term is computed as follows. Let  $\Delta K_i^{(min)}$  and  $\Delta K_i^{(max)}$  be respectively the minimum and maximum values that the degradation of performance  $K_i$  can assume with different values of parasitics.

$$f_{co} = \sum_{i=1}^{N_k} C_i$$

where  $C_i$  is given by

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

$\rho_c$  is the ratio between the maximum and minimum value of the minimum-width unit-length substrate capacitance of interconnections on the available routing layers. If  $S_r \gg 1$  then the values of  $C_i$  for feasible and infeasible placements differ by at least one order of magnitude. In our implementation,  $S_r = 10$ .

In case deterministic values for  $\mathbf{p}^{(\max)}$  are not available, the degradation variance  $\sigma^2(\Delta K_i)$  is computed with the model (12), and then compared with specification  $\overline{\sigma^2(\Delta K_i)}$ , using equation (15). The added measure of violation is then treated like any other performance violation and used to drive directly the annealing algorithm.

- $\alpha_{wl}, \alpha_a, \alpha_{ov}, \alpha_{we}, \alpha_{sy}, \alpha_{ma}$  and  $\alpha_{co}$  are non-negative weights. Their initial default values are adjusted dynamically during the algorithm using heuristics so that, at the beginning of the annealing, area and wire length dominate in the expression of the cost function, then their importance decreases progressively, until at low temperatures overlaps, symmetries, and constraint violations become dominating.

#### Abutment and control of junction capacitances

Device abutment during placement is useful to reduce interconnect and junction capacitances, and to obtain substantial gain in area. It can also be used to merge the diffusion regions of MOS transistors or of other components, such as capacitors, BJTs etc. In PUPPY-A, abutment is obtained in two different ways. The first is by dynamic device abutment (similar to the approach in KOAN [22]), performed by PUPPY-A during the annealing algorithm. In PUPPY-A, dynamic abutment is driven by parasitic constraints, as well as by area and wiring considerations. Instead of randomly choosing the devices to merge, the algorithm operates first on the nets whose parasitics are critical for performance constraints. The second is through the *stack generator* LDO, which efficiently builds stacks containing transistors all with the same width.

LDO [34] implements stacks of folded or interleaved MOS transistors sharing their drain and source diffusions. Dense layouts can be achieved with this approach, the junction capacitances associated with shared diffusions being minimized. Moreover, matching between transistors decomposed into elements stacked together is usually good, in particular if the elements are interleaved. Because of the regularity of these structures, routing is usually dense with this layout style.

The target of the stack generator can be summarized as follows:

1. obtain maximally compact stacks, so that the area occupied by the devices is minimum;
2. keep all critical capacitances at their minimum value by exploiting the abutment of source/drain diffusion areas;
3. provide control over device matching, so that critically matched devices can be decomposed into interleaved elements, and common-centroid structures are obtained when symmetry constraints are enforced;
4. provide control over net length, by conveniently distributing the elements within the stacks.

LDO is based on an algorithm exploiting the equivalence between stack generation and path partitioning in the circuit graph. The algorithm is guaranteed to find all optimum stacked configurations, according to an optimality criterion defined by a cost function, which takes into account parasitic criticality, matching constraints and device area. The stack generation algorithm is based on a two-phase approach, working on the *circuit graph*, i.e. a graph whose nodes are circuit nets, and whose edges are MOS transistors. In the first phase, a dynamic programming procedure generates all possible paths in the circuit graphs, namely in

the connected subgraphs whose nodes have no more than two adjacent edges. The second phase explores the compatibility between all paths. By solving a clique problem, an optimum set of paths is selected, which minimizes the cost function and contains all the transistors of the circuit. More details on the algorithm and its implementation can be found in [34].

In LDO, not only are symmetries fully taken into account, but they have proved effective to reduce the computational complexity by limiting the size of the search space, while preserving the admissibility of the algorithm (i.e. the optimum solution is always found). In practice, the higher the number of symmetry constraints, the faster the algorithm runs.

By abutting elemental transistors into one stack, their source/drain regions are merged, thus reducing effectively their junction capacitances. The cost function driving LDO tries to minimize the most critical capacitances, according to the tightness of their bounds. Junction capacitances are the only parasitics that can be directly controlled by LDO, because they are directly influenced by the shape and inner organization of the stacks. Routing parasitics, such as interconnect stray resistances and capacitances, can be controlled effectively only after the placement phase. This limitation can be overcome by simultaneously generating and placing the stacks. This has been achieved by means of an annealing move-set extension, to include a move called "alternative solution swap", which selects randomly a module in the circuit, and swaps it with one of its alternative implementations found by LDO. The criterion whether to accept the move is based on the usual annealing scheme [33].

#### Example

Consider the clocked comparator `COMPL`. Many possible stack implementations exist for this circuit. Two of such possible solutions are shown in Fig.4. All transistors have been grouped in four sub-circuits, according to their channel widths, their matching requirements and bulk nets. Only transistors belonging to the same sub-circuit can belong to the same stack. The two solutions only differ by the implementation of the stack containing the input differential pair. In the first realization they are interleaved in a common-centroid pattern, which minimizes device mismatch, but usually requires a considerable area overhead, due to the complex routing required. The second solution is symmetric, but without the common-centroid structure. The choice between such alternative realizations is left to the user or it can be made automatically during the placement phase on the ground of area and routing considerations. In both solutions, critical nets 55, 56, 15, 16, whose capacitance toward the substrate strongly influences the comparator speed, have been kept in internal positions when possible. Their capacitances are reported in Table 2. In both cases stack abutment yielded a reduction of net capacitance. Such a reduction can be exploited to improve the flexibility of the routing stage. For example, consider nets 55 and 56. Abutment allowed each of them

to be reduced by more than 6.6fF, which in our process is the capacitance of a 136 $\mu$ m-long minimum-width metal-1 wire. Therefore the router is allowed to draw longer wires for the sensitive nets, thus increasing the success rate and the robustness of the entire layout synthesis.

These capacitance values constitute new nominal values and better lower limits, and can be used to compute a new set of bounds. By using these values:

$$\begin{aligned} C_{15}^{(min)} &= C_{15}^{(nom)} = C_{16}^{(min)} = C_{16}^{(nom)} = 34.4fF \\ C_{55}^{(min)} &= C_{55}^{(nom)} = C_{56}^{(min)} = C_{56}^{(nom)} = 6.6fF \\ \max C &= 100fF \\ \min R &= 0 \\ \max R &= 50\Omega \end{aligned} \quad (16)$$

we obtain the following arrays:

$$\mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 5.5ns \\ 0.0 \\ 0.0 \end{bmatrix} \quad \overline{\Delta \mathbf{K}} = \begin{bmatrix} 1.5ns \\ 1mV \\ 1mV \end{bmatrix}$$

Here the delay degradation, due to the insertion of junction capacitances, is apparent. The next set of bounds found by PARCAR is the following:

$$\mathbf{p}^{(b)} = \begin{bmatrix} 67.1fF \\ 67.1fF \\ 48.9fF \\ 48.9fF \\ 1.0\Omega \\ 7.4\Omega \\ 7.4\Omega \\ 7.4\Omega \\ 7.5\Omega \\ 19.9\Omega \\ 49.5\Omega \end{bmatrix} \quad (17)$$

Notice that all bounds on critical capacitances have been lowered, because the degradation allowed to delay is smaller than in the previous step. In fact half of the degradation allowed at the beginning of the layout design has been introduced by junction capacitances alone, and the remaining half will be available to the remaining tools (i.e. placement and routing tools). The placement of Fig.5 was obtained with the set of bounds (17). After placement, estimates of the minimum values of all critical parasitics can be drawn, taking into account the junction capacitances of all terminals and the estimated minimum length of interconnections between terminals:

$$\mathbf{p}^{(0)} = \mathbf{p}^{(min)} = \begin{bmatrix} 10.1fF \\ 10.1fF \\ 51.0fF \\ 51.0fF \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad (18)$$

## V. ROUTING WITH ANALOG CONSTRAINTS

### Channel Routing

In the channel router ART [24], the two-layer gridless channel-routing problem [47] is represented by a *vertical-constraint graph (VCG)* whose nodes correspond to the horizontal segments of a net or subnet. An undirected edge links two nodes if the associated segments have a common horizontal span. A directed edge links two nodes if one segment has to be placed above the other because of pin constraints. The weight of an edge is the minimum distance between the center lines of two adjacent segments. Hence the channel routing problem is formulated as the problem of directing all the undirected edges so as to minimize the longest directed path in the VCG. The length of such path corresponds to the channel width. Over-constraints can be solved by assigning to each net more than one node in the VCG. However, with this approach we introduce additional capacitive couplings due to the wire jogs. In the current implementation of ART, one VCG node is supported for each net.

In ART, all parasitic bounds are mapped into constraints for the VCG. Within a channel, nets provide two different contributions to cross-coupling capacitance: crossover capacitances between overlapped orthogonal wire segments, and capacitances between segments running parallel to each other. Both depend on the distribution of terminals along the channel edges. Unavoidable crossovers can be determined directly on the ground of the terminal positions. If such a crossover is detected, it introduces a lower bound for the cross-coupling capacitance between the corresponding nets. The coupling between horizontal adjacent edges is controlled by their minimum separation, and therefore it is proportional to the weight of the corresponding edge in the VCG. This contribution can be theoretically reduced at will by inserting sufficient space, or by exploiting the shielding effect due to other wires. Shielding nets can be inserted on purpose, if the presence of a further wire segment in the channel is more convenient in terms of area than extra spacing. In ART, this is automatically carried out by adding a new node and edges to the VCG.

Perfect mirror symmetry can be achieved when symmetric nets are restricted to different sides of the symmetry axis (i.e. they don't cross). If the horizontal spans of a pair of symmetric nets intersect the symmetry axis, perfect mirror symmetry cannot be achieved. However, good parasitic matching can be obtained between the nets with the technique illustrated in Fig.6. A "connector" allows two symmetric segments to cross over the axis. Resistances and capacitances of the two nets are matched, because for each one the connector introduces the same interconnect length, the same number of corners on each layer, and the same number of vias. Only coupling capacitances with other nets running close to the connector will suffer slight asymmetries.

### Area Routing

ROAD [31] is a maze router based on the A\* algorithm

[48], using a relative grid with dynamic allocation. For each net, the path found by the maze router is the one of minimum length. If a cost function is defined on the edges of the grid, the path found is the one minimizing the integral of the cost function. In ROAD, the cost function is a weighted sum of several non-homogeneous items. Let  $\mathcal{N}$  be the set of all nets. On a given grid edge  $x$  with length  $L(x)$ , on layer  $l$ , the cost function for a net  $N \in \mathcal{N}$  has the following form:

$$F(x) = L(x) \cdot \left( 1 + \frac{Cr(x)}{Cr_0} + w_R \frac{R_u(l)}{R_0} + w_{C_u} \frac{C_u(l)}{C_0} + \frac{1}{C_P} \sum_{n \in \mathcal{N} - \{N\}} w_{C_n} C_n(x) \right) \quad (19)$$

where:

- $Cr(x)$  is a measure of local area crowding. It is computed in a simplified form, by giving over-congested areas steep cost function "hills", which prevent future wires from crossing these areas. Area crowding  $Cr(x)$  is given by

$$Cr(x) = \begin{cases} 0, & \text{if } R \leq 1 \\ Cr_{max} & \text{if } R > 1 \end{cases} \quad (20)$$

where  $Cr_{max}$  is a large constant (the height of the "hills"), and  $R$  is the ratio between the needed room for the new wire which has to be built, and the room available on the sides of edge  $x$ :

$$R = \frac{\text{needed room}}{\text{available room}}.$$

- $R_u(l)$  is the resistance of a minimum-width unit-length wire segment on layer  $l$ .
- $C_u(l)$  is the capacitance to bulk of a minimum-width unit-length wire segment on layer  $l$ . The model is described in Appendix.
- $C_n(x)$  is the capacitance between a unit-length wire segment located across edge  $x$  and the wire implementing net  $n$ . The model is described in Appendix.
- $w_{R_u}$ ,  $w_{C_u}$  and  $w_{C_n}$  are weights regulating the relative importance of each item.
- $Cr_0$ ,  $R_0$ ,  $C_0$ ,  $C_P$  are reference parameters providing dimensional homogeneity to the addenda and a meaning to their comparison.

Weights provide an efficient way to limit the magnitude of critical parasitics. Performance sensitivities to parasitics are used to generate the weights for the cost function driving the area router. The contribution of a parasitic to performance degradation is proportional to the sensitivity and inversely proportional to the maximum variation range allowed to that performance. The weight  $w_j$  associated to parasitic  $p_j$  is defined as follows:

$$w_j = \sum_{i=1}^{N_k} \left( \frac{S_{i,j}^-}{\Delta K_i^-} + \frac{S_{i,j}^+}{\Delta K_i^+} \right) P_0$$

where  $P_0$  is a normalization factor, such that, if sensitivities are not all zero, at least one weight is set to 1, and the

others are all between 0 and 1. The dimensional unit of  $P_0(\Omega, V, F, \dots)$  depends on the parasitic type. With this definition, each item in (19) can be interpreted as the contribution to performance degradation due to one of the parasitics introduced by the wire segment routed along edge  $x$  of the grid.

The routing schedule is determined with a set of heuristic rules set up and tuned with experimental tests. The higher the number of constraints on a net, the higher is its priority. We define a number of *properties* that a net can have, for instance symmetry, belonging to the class of supply nets, of clocks etc. The priority of net  $n$  is given by the following expression:

$$Pr(n) = \sum_j a_j \text{Prop}_j(n) + \sum_{i=1}^{N_p} a_{p_i} w_{p_i},$$

where  $\text{Prop}_j(n)$  is 1 if net  $n$  has the  $j$ -th property, and 0 otherwise. Parameters  $w_{p_i}$  are the same parasitic weights used to define the cost function (19), while  $a_j$  and  $a_{p_i}$  are *priority weights* expressing the importance of each property. Priority weights are assigned in such a way that maximum priority is given to symmetric nets, followed by supply nets and then by nets with tight electrical requirements. The most difficult nets are routed first, and the unrouted ones are less and less critical as the circuit crowding increases. If two nets have the same priority, the shorter one is routed first.

After performing the weight-driven routing, parasitics are extracted and performance degradation is estimated and compared with its specifications. If constraints (6) are not met, the weights of the most sensitive parasitics are raised and routing is repeated. When the weights of all sensitive parasitics hit their maximum value (that is 1), iterations stop. This means that even considering maximum criticality for the sensitive parasitics, routing is not possible on the given placement, without constraint violations. In this case, the circuit placement needs to be generated again, using a wider range of variation for the detected sensitive parasitics.

### Symmetries

ROAD is able to find symmetric paths for differential signals with symmetric placements, even in the presence of a non-symmetric distribution of terminals. The algorithm, described in detail in [31], is illustrated in Fig.7. Let us assume without loss of generality that the symmetry axis is vertical and it splits the circuit into a left half and a right half. If the placement is not perfectly symmetric, we consider the outline determined by the union of real obstacles and *virtual obstacles* obtained by mirroring each obstacle with respect to the symmetry axis. First every net is built considering only the terminals located on the left side of the symmetry axis or on the border of the wiring space, that is not contained by any virtual image of an obstacle. The wire segments defined in this way are called *left-side segments*. Then, each left-side segment is mirrored with respect to the symmetry axis. Next, the routing is extended

to cover the portions of area occupied by virtual obstacles, but not by real obstacles. The segments whose existence is not required for the full net connectivity are pruned. Only those branches of non-symmetric nets should be pruned, that don't cross or run close to symmetric nets.

### Electrostatic Shields

Decoupling based only on wire spacing can increase excessively area, and this can be avoided by inserting wire stubs, connected to a virtual ground, shielding critically coupled wires. Shields are built after all wires have been routed. Given two wires to be decoupled, first a grid node between each pair of parallel segments of the two wires is found, or generated by dynamic grid allocation. On each of these node, area congestion is computed with expression (20), and a terminal is defined wherever congestion is sufficiently low, i.e. where  $Cr(x) < Cr_{max}$ . Next, a new wire is routed through all these terminals, and connected (with null weight on resistive constraints) to the proper ground node. If local congestion doesn't allow to create a suitable shield, then we pass to the re-route phase as described above.

### Example

Consider the clocked comparator COMPL. After the placement step, the nominal values of all parasitics have been updated as shown in (18). With these new nominal values, the least capacitive interconnect among the terminals of the critical nets would give (by simulation) a total delay of 5.9ns:

$$\mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 5.9ns \\ 0.0 \\ 0.0 \end{bmatrix} \quad \overline{\Delta\mathbf{K}} = \begin{bmatrix} 1.1ns \\ 1mV \\ 1mV \end{bmatrix}$$

With the high-level constraints (13), now only 1.5ns of delay degradation are allowed to the router, of which 0.4ns have been recognized as being unavoidable with this placement. Therefore tight bounds will have to be enforced by the router. In fact PARCAR now requires the following bounds:

$$\mathbf{p}^{(b)} = \begin{bmatrix} 76.5fF \\ 76.5fF \\ 39.3fF \\ 39.3fF \\ 1.0\Omega \\ 7.4\Omega \\ 7.4\Omega \\ 7.4\Omega \\ 7.5\Omega \\ 19.9\Omega \\ 49.5\Omega \end{bmatrix}$$

Comparing the capacitive bounds with the junction capacitances (16) computed after module generation, it is evident that 42.5fF are available for routing nets 15 and 16, and 32.7fF are available for nets 55 and 56. The layout routed by ROAD is shown in Fig.8.

Extraction results for capacitances are the following:

$$\begin{aligned} C_{15} &= 70.3fF \\ C_{16} &= 70.4fF \\ C_{55} &= 20.6fF \\ C_{56} &= 18.0fF \end{aligned}$$

Simulation results after extraction of the routed layout give a delay of  $6.5ns$  and an offset of  $756\mu V$ .

## VI. COMPACTION

SPARCS-A [49] is a mono-dimensional constraint-graph (CG) longest-path compactor, implementing algorithms to enforce symmetry and parasitic constraints. The role of compaction in the constraint-driven approach is important for two reasons:

1. Constraints enforced by the previous layout steps, such as parasitic bounds, symmetries, and shields, should not be disrupted for the sake of area minimization. The compactor must be able to respect and if necessary to enforce such constraints.
2. The compactor can recover design-rule errors and constraint violations. Hence the requirements on placement and routing in terms of constraint enforcement can be relaxed. Since compaction has generally higher computational efficiency than routing, the overall CPU cost of layout design can be substantially reduced. Thus, in addition to reducing chip area, compaction also improves the efficiency and robustness of the entire analog synthesis process by permitting the use of more aggressive techniques during placement and routing.

The algorithm implemented in SPARCS-A takes advantage of the high speed of the CG-based technique to provide a good starting point to a Linear Programming (LP) solver. Mono-dimensional compaction is iterated alternatively in the two orthogonal directions, until no area improvement is achieved. The algorithm used in each iteration is the following:

1. With the CG technique, solve the spacing problem without symmetry constraints.
2. Use the simplex linear programming algorithm to solve the symmetry constraints, using the CG solution as initial starting point.
3. Round off the coordinates of the elements laying out of the critical path
4. Verify that all constraints are satisfied

Using the CG solution as starting point is key to a significant speed-up in the solution of the linear problem. Compared to previous approaches [23] solving compaction using an LP solver, this algorithm represents a substantial improvement. In fact in our case the LP solver starts from a feasible configuration which is already close to the final solution. The range of cases that can be managed with acceptable computational complexity is therefore significantly broadened [50]. Control over cross-coupling capacitances is enforced by modifying the constraint-graph before computing the longest path. Proper distances between

parallel interconnection edges are kept to maintain cross-coupling capacitances below their bounds. This is achieved by employing a heuristic which adds extra spacing between wire segments, based on the need for decoupling and on their length, which has a direct impact on the overall area. The procedure implementing this heuristic is the following.

### procedure `modify-graph`

```

/* Purpose: Add constraints to improve capacitive decoupling
/* Since this procedure is called only if some performance violation
/* has been found, we know that at least one bound has been exceeded.
for each cross-coupling  $C_j$  such that  $C_j > C_j^b$ :
  let  $\delta_j =$  current min. distance between any two parallel segments
  contributing to  $C_j$ 
   $\delta_j = \delta_j + mindist(C_j, C_j^b)$ 
  for each pair  $P_i$  of parallel segments:
    let  $d_i =$  current minimum distance between the segments of pair  $P_i$ ;
    if  $d_i < \delta_j$  then
      Add constraint to graph requiring  $d_i \geq \delta_j$  between the segments
  end
end

```

$C_j$  indicates the  $j$ -th cross-coupling capacitance, and  $C_j^b$  is the bound on its maximum value. Function  $mindist(C_j, C_j^b)$ , which depends on the model used for capacitances (see Appendix), returns the minimum distance increment to add between parallel segments of the  $j$ -th pair of wires, to reduce their cross-coupling capacitance from  $C_j$  to  $C_j^b$ . For each cross-coupling  $C_j$  exceeding its bound  $C_j^b$ ,  $\delta_j$  is the minimum distance to be kept between parallel segments of the  $j$ -th pair of wires. The distance increment is a function of the parasitic bound violation: the bigger the violation, the wider the extra spacing added. Notice that in procedure `modify-graph`, spacing is added not only between the nearest segments, but also between all the segment pairs whose distance is less than  $\delta_j$ .

The spacing step implemented by procedure `modify-graph` can introduce over-constraints making the graph unsolvable. An example where this situation might occur is illustrated in Fig.9. Two wire segments are connected to terminals A and B, whose relative position is fixed with respect to the instance of a sub-cell. An over-constraint, due to a positive loop in the CG, is generated if the spacing required between the segments is  $\delta_j > D - W_1 - W_2$ . When a positive loop is detected, a *pruning* procedure is invoked, which removes the newly-added spacing constraints contained in the positive-weight loops. In such situation, the task of decoupling the two nets is left to the remaining segment pairs. If a feasible solution involving the remaining segment pairs does not exist, an error is reported because the constraint cannot be met.

One of the main advantages of the longest-path compaction algorithm is that right after each compaction step it provides the exact value of the minimum layout pitch. This can be exploited to add geometric constraints together with electrical performance specifications. Let us consider, without loss of generality, a horizontal compaction step. The pitch  $W$  of the longest path can be checked against the maximum size  $W_{max}$  allowed to the circuit width. If it is smaller, the difference between them is the maximum amount by which the vertical parallel wire segments be-

longing to the longest path itself can be brought apart from each other. Otherwise, all such pairs of wire segments must be kept at their minimum distance. This corresponds to an additional constraint:

$$\sum_i \delta_i \leq \min(0, W_{max} - W_{lp}) \quad (21)$$

where the sum is extended to all the vertical segment pairs laying on the longest path. Aspect ratio constraints can be reduced to absolute-size constraints by considering, at each step, the pitch of the layout in the orthogonal direction as fixed.

#### Example

Consider once more the clocked comparator COMPL. As shown at the end of Section V, after routing the problem instances are the following:

$$\mathbf{p}^{(0)} = \begin{bmatrix} 70.3fF \\ 70.4fF \\ 20.6fF \\ 18.0fF \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 6.2ns \\ 756\mu V \\ -756\mu V \end{bmatrix}$$

$$\overline{\Delta \mathbf{K}} = \begin{bmatrix} 0.8ns \\ 244\mu V \\ -244\mu V \end{bmatrix}$$

The solution to the constraint-generation problem is

$$\mathbf{p}^{(b)} = \begin{bmatrix} 87.7fF \\ 87.8fF \\ 32.8fF \\ 28.4fF \\ 1.0\Omega \\ 1.0\Omega \\ 1.0\Omega \\ 1.0\Omega \\ 1.0\Omega \\ 5.2\Omega \\ 49.3\Omega \end{bmatrix}$$

Notice that now resistive mismatch has become critical because of the shrunk margin allowed to offset degradation. Of the capacitive parasitics,  $C_{55}$  and  $C_{56}$  have been recognized as more critical than the others, and their bounds have been further tightened with respect to their previous values used to drive the router. Other bounds on  $C_{15}$  and  $C_{16}$  have been relaxed as a consequence. The layout produced with this set of bounds is shown in Fig.10. Capacitive extraction from this layout yields the following values:

$$\begin{aligned} C_{15} &= 73.9 \\ C_{16} &= 75.2 \\ C_{55} &= 18.8 \\ C_{56} &= 17.2 \end{aligned}$$

Simulation showed that in the compacted layout performance specifications were met with an offset of  $743\mu V$  and a delay of  $6.7ns$ .

## VII. RESULTS

All the tools described in this paper have been implemented within the OCTTOOLS framework of the University of California at Berkeley. This has allowed us to test the described algorithms, and to validate the methodological approach on a large set of test circuits. All sensitivity computation and simulations have been done using SPICE-3 [38].

This section reports a few circuit examples to illustrate the methodology and the results which can be achieved.

#### Example: two-stage CMOS opamp

Consider the two-stage CMOS opamp shown in Fig.11, with specification constraints on offset  $V_{off}$  and on unity-gain bandwidth  $\omega_0$ :

$$\begin{aligned} |V_{off}| &< 2.6mV \\ \omega_0 &> 6.5MHz \end{aligned}$$

Simulation results confirm that with no parasitics the nominal values of systematic offset and bandwidth would be  $V_{off}^{(0)} = 2.4mV$ , and  $\omega_0^{(0)} = 6.6MHz$  respectively. Hence:

$$\mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 2.4mV \\ -2.4mV \\ -6.6MHz \end{bmatrix} \quad \overline{\Delta \mathbf{K}} = \begin{bmatrix} 0.2mV \\ 5.0mV \\ 0.1MHz \end{bmatrix}$$

For sake of clarity only a portion of the circuit in Fig.11 is quantitatively analyzed in terms of the effects on performance by interconnect parasitics. In Fig.12 the input differential pair and its active load are shown. Three different solutions found by LDO are shown in Fig.13. All meet matching and symmetry requirements, have the same values for all critical junction capacitances, and require the same area for active devices, i.e. for LDO the costs of the three implementations are the same. According to the notation defined in Table 1, the critical parasitic array and the matrix of sensitivities are:

$$\mathbf{p} = \begin{bmatrix} R_{S-1} \\ R_{S-2} \\ R_{S-2,1} \\ R_{S-3,4} \\ V_{t-2,1} \\ V_{t-3,4} \\ C_{6,9} \\ C_6 \\ C_9 \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 0.0 & 0.0 & 0.46KHz/\Omega \\ 0.0 & 0.0 & 0.51KHz/\Omega \\ 0.020mV/\Omega & 0.020mV/\Omega & 0.0 \\ 0.013mV/\Omega & 0.013mV/\Omega & 0.44KHz/\Omega \\ 5.00 \times 10^{-4} & 5.00 \times 10^{-4} & 0.0 \\ 1.25 \times 10^{-4} & 1.25 \times 10^{-4} & 0.0 \\ 0.0 & 0.0 & 1.21KHz/fF \\ 0.0 & 0.0 & 1.03KHz/fF \\ 0.0 & 0.0 & 0.84KHz/fF \end{bmatrix}^T$$

Notice that offset sensitivities are not null only w.r.t. mismatch parameters. Consider now the three alternative implementations shown in Fig.13. Realization (a) requires the smallest routing area for nets 6 and 9. This implies low interconnect resistances and capacitances and therefore low  $\omega_0$  degradation. Estimations of parasitics would yield a  $\Delta\omega_0$  of  $-74KHz$  and a  $\Delta V_{off}$  of  $0.3mV$ . Clearly only one specification can be met with such a configuration, unless the specification on offset is relaxed. This is mainly due to the role played by the threshold voltage mismatch. In realization (c) both the differential pair and the active load are

tightly interleaved, and therefore the threshold voltage mismatch is minimized, though at the expenses of capacitive cross-coupling and of substrate capacitance of nets 6 and 9. In fact  $\omega_0$  exceeds the specifications (-173.98 KHz), while  $V_{off}$  becomes acceptable (150.33  $\mu V$ ). If both tight performance constraints are specified simultaneously, a trade-off configuration must be chosen. For instance, in realization (b) the differential pair is interleaved in a common-centroid pattern, while the active load is implemented in a simpler way. With this configuration, both constraints are satisfied. In fact  $\Delta V_{off}$  and  $\Delta\omega_0$  are 148.0  $\mu V$  and 97.2 KHz, respectively. Notice that this particular layout cannot be found with any tool relying only on automatic abutment during placement, because of the interleaved pattern in the differential pair. Nor could it be generated with standard module generators [51], unless a detailed knowledge of the circuit structure was known a priori.

*Example: FASTCOMP*

Fig.14 shows the schematic of a clocked comparator named FASTCOMP. For this circuit we consider specifications on voltage offset and switching speed. The nominal values are

$$\begin{aligned} V_{off} &= 0.0mV \\ \tau_D(H \rightarrow L) &= 2.42ns \\ \tau_D(L \rightarrow H) &= 2.49ns \end{aligned}$$

The constraint specifications are

$$\begin{aligned} |V_{off}| &\leq 2.0mV \\ |\Delta\tau_D(H \rightarrow L)| &\leq 0.25ns \\ |\Delta\tau_D(L \rightarrow H)| &\leq 0.25ns \end{aligned}$$

Therefore

$$\mathbf{K} = \begin{bmatrix} V_{off} \\ -V_{off} \\ \tau_D(H \rightarrow L) \\ -\tau_D(H \rightarrow L) \\ \tau_D(L \rightarrow H) \\ -\tau_D(L \rightarrow H) \end{bmatrix} \quad \mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 0.0 \\ 0.0 \\ 2.42ns \\ -2.42ns \\ 2.49ns \\ -2.49ns \end{bmatrix}$$

$$\overline{\Delta\mathbf{K}} = \begin{bmatrix} 2mV \\ 2mV \\ 0.25ns \\ 0.25ns \\ 0.25ns \\ 0.25ns \end{bmatrix}$$

Table 3 shows some of the most critical parasitic constraints found by PARCAR. As expected, the main contribution to voltage offset is due to parasitic resistances responsible for source degeneration of the input pair. The input source followers (MP10-11 and MP8-9) are less critical than the high-gain pairs (MN3-4, MN1-2 and MP2-3).

The complete layout of FASTCOMP is shown in Fig.15. Fig.16 shows two details of the layout area highlighted in Fig.15, respectively with and without parasitic and topological constraint enforcement. In the right-hand side example, large capacitive couplings between critical nets are

clearly visible. In particular, a considerable mismatch is present between nets 7 and 8. The capacitance of nets 3,4,9,10 is large, thus slowing down the signal path. These capacitances are much smaller in the example shown in the left-hand side. Notice that relatively large cross-couplings between nets 3,4 and 9 were accepted due to their low criticality. A performance comparison of both the constrained and the unconstrained layouts is summarized in Table 4. Table 5 lists the CPU times required on a DEC-station 5000/240 for each layout phase.

*Example: MPH*

Fig.17 shows the schematic of a micro-power amplifier. This is an example that shows how the layout methodology described in this paper fits also tight constraint specifications on relatively large circuits. The nominal performance values for this circuit are the following:

$$\begin{aligned} V_{dd} &= 1.5V \\ \omega_0 &= 6.0MHz \\ A_v &= 120dB \\ \phi_M &= 60^\circ \end{aligned}$$

The following constraints have been specified:

$$\begin{aligned} |\Delta V_{dd}| &\leq 150mV \\ \Delta\omega_0 &\geq -100KHz \\ \Delta A_v &\geq -0.1dB \\ |\Delta\phi_M| &\leq 10^\circ \end{aligned}$$

Therefore

$$\mathbf{K} = \begin{bmatrix} V_{dd} \\ -V_{dd} \\ -\omega_0 \\ -A_v \\ \phi_M \\ -\phi_M \end{bmatrix} \quad \mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 1.5V \\ -1.5V \\ -6.0MHz \\ -120dB \\ 60^\circ \\ -60^\circ \end{bmatrix} \quad \overline{\Delta\mathbf{K}} = \begin{bmatrix} 150mV \\ 150mV \\ 100KHz \\ 0.1dB \\ 10^\circ \\ 10^\circ \end{bmatrix}$$

The complete layout of MPH is shown in Fig.18.

Results for this layout are reported in Table 6, compared with the data from a hand-made implementation of the same circuit, made by an experienced designer. The comparison between the layouts shows the usefulness of the constraint-driven approach. Table 7 shows the CPU times required by each phase of the design, referred to a Decstation 5000/240. Due to the very large set of critical parasitics and the tightness of constraints, PARCAR required a considerably longer CPU time than with FASTCOMP and the previous examples.

Table 8 summarizes the results obtained with the tools described in this paper on a set of benchmarks of industrial strength. In each of these examples, all the performance specifications have been met. CPU times refer to a Decstation 5000/240.

## VIII. CONCLUSIONS

In this paper we have presented a constraint-driven methodology for the design of analog layout, supported by a set of specialized tools. The key points of the methodology can be summarized as follows:

- We apply a rigorous methodology to translate high-level performance specifications into the set of constraints that the tools are able to control. The constraint generation technique guarantees that if we can satisfy the low-level constraints, all high-level specifications will be met.
- At each step of the layout design the tools are able to enforce constraints on all low-level parameters of the circuit.
- Infeasibility is detected as soon as possible in the design flow. A quantitative analysis allows us to determine the causes of infeasibility and to address a re-design strategy.

The tools presented cover all the major steps of layout synthesis, namely placement, routing and compaction. The presence of a constraint-aware compactor allows the routing phase a more aggressive approach, thus improving the success rate and the robustness of the entire synthesis. All tools have been integrated in an environment where they share the data-base, the constraint representation, the parasitic models, and the performance analysis methods.

The impact of each layout step on the flexibility of the entire design flow has been analyzed in detail. The examples shown are benchmarks of industrial strength, and validate the effectiveness of our methodology.

#### ACKNOWLEDGMENTS

The authors would like to thank the students, researchers and professors of the Electronics Research Lab. of the Dept. of EECS of the University of California, for many discussions on the topics of this paper. This work could not have been possible if we had not been surrounded by such a stimulating human environment.

Some of the examples shown in this paper have been provided by Dr. Marco Gandini of CSELT Laboratories, Torino, Italy, Dr. John Cohn of IBM Corp., Essex Jct. VT, Prof. J. H. Huijsing and Dr. R. G. H. Eschauzier of Delft University of Technology, The Netherlands, whom we gratefully acknowledge.

#### APPENDIX

##### *Capacitive models for interconnections*

This appendix reports the capacitive models adopted to compute the stray capacitances of interconnections. In all our tools, the capacitive models described by [52] have been used. The dependence of the capacitance between a net and the substrate, or between two nets, is expressed as a polynomial in terms of wire widths and spacings. The coefficients are technology dependent, and can be computed, for each process, by accurate three-dimensional simulation as in [52], by interpolation on experimental measurement, or by solving the Laplace equation when geometries are sufficiently regular.

The capacitance between a unity-length wire segment and the substrate is given by:

$$C_u = k_0 + k_1 w$$

where  $w$  is the wire segment width.

The capacitance between unity-length parallel wire segments is given by:

$$C_{parallel} = k_0' + k_1' w_1 + k_2' w_2 + \frac{k_3'}{d} + \frac{k_4'}{d^2} \quad (22)$$

where  $d$  is the distance between the wire segments, and  $w_1, w_2$  are their respective widths. Model (22) holds for wires on the same layer or on different layer. In the former case, the model is symmetric, i.e.  $k_1' = k_2'$ , in the latter case  $k_1'$  and  $k_2'$  may differ.

The capacitance between orthogonal wire segments crossing each other is given by:

$$C_{cross} = k_0'' + k_1'' w_1 + k_2'' w_2 + k_3'' w_1 w_2 \quad (23)$$

where  $w_1$  and  $w_2$  are the widths of the two segments. The fringe effect is accounted for by the two linear terms, while the quadratic term corresponds to the parallel-plate contribution, proportional to the crossing area ( $w_1 w_2$ ).

#### REFERENCES

- [1] H. Y. Koh, C. H. Séquin and P. R. Gray, "OPASYN: A compiler for CMOS operational amplifiers", *IEEE Trans. on CAD*, vol. 9, n. 2, pp. 113-126, February 1990.
- [2] H. Onodera, H. Kanbara and K. Tamaru, "Operational Amplifier Compilation with Performance Optimization", in *Proc. IEEE CICC*, May 1989.
- [3] R. Harjani, R. A. Rutenbar and L. R. Carley, "OASYS. A Framework for Analog Circuit Synthesis", Research Report CMUCAD-89-65, Carnegie Mellon University, Pittsburgh, PA, Nov 1989.
- [4] H. Yaghtiel, A. Sangiovanni-Vincentelli and P. R. Gray, "A Methodology for Automated Layout of Switched-Capacitor Filters", in *Proc. IEEE ICCAD*, pp. 444-447, 1986.
- [5] Y. Therasse, L. Reynders, R. Lannoo and B. Dupont, "A Switched-Capacitor Filter Compiler", *VLSI System Design*, pp. 85-88, Sept. 1987.
- [6] J. Assael, P. Senn and M. S. Tawfik, "A Switched-Capacitor Filter Silicon Compiler", *IEEE Journal of Solid State Circuits*, vol. 23, n.1, pp. 166-174, February 1988.
- [7] P. E. Allen and P. R. Barton, "A Silicon Compiler for Successive Approximation A/D and D/A Converters", in *Proc. IEEE CICC*, pp. 552-555, 1986.
- [8] G. Jusuf, P. R. Gray and A. Sangiovanni-Vincentelli, "CADICS - Cyclic Analog-To-Digital Converter Synthesis", in *Proc. IEEE ICCAD*, pp. 286-289, November 1990.
- [9] H. Chang, E. Liu, R. Neff, E. Felt, E. Malavasi, E. Charbon, A. Sangiovanni-Vincentelli and P. R. Gray, "Top-Down, Constraint-Driven Methodology Based Generation of n-bit Interpolative Current Source D/A Converters", in *Proc. IEEE CICC*, pp. 369-372, May 1994.
- [10] G. Winner et al., "Analogue Macrocell Assembler", *VLSI System Design*, pp. 68-71, May 1987.
- [11] 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 CICC*, pp. 72-78, 1985.
- [12] T. Pleterssek et al., "High-Performance Designs with CMOS Analogue Standard Cells", *IEEE Journal of Solid State Circuits*, vol. 21, n.2, pp. 215-222, April 1986.
- [13] L. D. Smith et al., "A CMOS-Based Analog Standard Cell Product Family", *IEEE Journal of Solid State Circuits*, vol. 24, n. 2, pp. 370-379, April 1989.
- [14] 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.
- [15] M. G. R. Degrauwe et al., "Towards an Analog System Design Environment", *IEEE Journal of Solid State Circuits*, vol. 24, n. 3, pp. 659-671, June 1989.
- [16] M. Kayal, S. Pigué, M. Declercq and B. Hochet, "SALIM: A Layout Generator Tool for Analog ICs", in *Proc. IEEE CICC*, pp. 751-754, May 1988.

- [17] M. Mogaki, N. Kato, Y. Chikami, N. Yamada and Y. Kobayashi, "LADIES: An Automatic Layout system for Analog LSI's", in *Proc. IEEE ICCAD*, pp. 450-453, November 1989.
- [18] F. M. Turkey and E. E. Perry, "BLADES: An A.I. Approach to Analog Circuit Design", *IEEE Trans. on CAD*, vol. CAD-8, n. 6, pp. 680-692, June 1989.
- [19] R. S. Gyurcsik and J.-C. Jeon, "A Generalized Approach to Routing Mixed Analog and Digital Signal Nets in a Channel", *IEEE Journal of Solid State Circuits*, vol. 24, n. 2, pp. 436-442, Apr 1989.
- [20] I. Harada, H. Kitazawa and T. Kaneko, "A Routing System for Mixed A/D Standard Cell LSI's", in *Proc. IEEE ICCAD*, pp. 378-381, November 1990.
- [21] 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.
- [22] 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.
- [23] R. Okuda, T. Sato, H. Onodera and K. Tamaru, "An Efficient Algorithm for Layout Compaction Problem with Symmetry Constraints", in *Proc. IEEE ICCAD*, pp. 148-151, November 1989.
- [24] 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.
- [25] E. Malvasi, U. Choudhury and A. Sangiovanni-Vincentelli, "A Routing Methodology for Analog Integrated Circuits", in *Proc. IEEE ICCAD*, pp. 202-205, November 1990.
- [26] S. K. Hong and P. E. Allen, "Performance driven analog layout compiler", in *Proc. IEEE Int. Symposium on Circuits and Systems*, pp. 835-838, 1990.
- [27] E. Charbon, E. Malvasi, U. Choudhury, A. Casotto and A. Sangiovanni-Vincentelli, "A Constraint-Driven Placement Methodology for Analog Integrated Circuits", in *Proc. IEEE CICC*, pp. 2821-2824, May 1992.
- [28] U. Choudhury and A. Sangiovanni-Vincentelli, "Constraint Generation for Routing Analog Circuits", in *Proc. IEEE/ACM DAC*, pp. 561-566, June 1990.
- [29] G. Gad-El-Karim and R. S. Gyurcsik, "Use of Performance Sensitivities in Analog Cell Layout", in *Proc. IEEE Int. Symposium on Circuits and Systems*, June 1991.
- [30] H. Chang, A. Sangiovanni-Vincentelli, F. Balarin, E. Charbon, U. Choudhury, G. Jusuf, E. Liu, E. Malvasi, R. Neff and P. Gray, "A Top-down, Constraint-Driven Design Methodology for Analog Integrated Circuits", in *Proc. IEEE CICC*, pp. 841-846, May 1992.
- [31] E. Malvasi and A. Sangiovanni-Vincentelli, "Area Routing for Analog Layout", *IEEE Trans. on CAD*, vol. 12, n. 8, pp. 1186-1197, August 1993.
- [32] E. Felt, E. Malvasi, E. Charbon, R. Totaro and A. Sangiovanni-Vincentelli, "Performance-Driven Compaction for Analog Integrated Circuits", in *Proc. IEEE CICC*, pp. 1731-1735, May 1993.
- [33] E. Charbon, E. Malvasi, D. Pandini and A. Sangiovanni-Vincentelli, "Simultaneous Placement and Module Optimization of Analog IC's", in *Proc. IEEE/ACM DAC*, pp. 31-35, June 1994.
- [34] E. Malvasi and D. Pandini, "Optimum CMOS Stack Generation with Analog Constraints", *IEEE Trans. on CAD/ICAS*, vol. 14, n. 1, pp. 107-122, January 1995.
- [35] 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 Trans. on CAD*, vol. 7, n. 4, pp. 501-519, April 1988.
- [36] J.-M. Shyu, "Performance Optimization of Integrated Circuits", Memorandum UCB/ERL M88/74, University of California at Berkeley, Nov 1988.
- [37] S. W. Director and R. A. Rohrer, "The generalized adjoint network and network sensitivities", *IEEE Trans. on Circuit Theory*, vol. 16, pp. 318-323, August 1969.
- [38] U. Choudhury, "Sensitivity Computation in SPICE3", M.S. Thesis, University of California at Berkeley, 1988.
- [39] 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.
- [40] M. J. M. Pelgrom, A. C. J. Duinmaijer and A. P. G. Welbers, "Matching Properties of MOS Transistors", *IEEE Journal of Solid State Circuits*, vol. 24, pp. 1433-1440, October 1989.
- [41] C. Guardiani, A. Tomasini, J. Benkoski, M. Quarantelli and P. Gubian, "Applying a submicron mismatch model to practical IC design", in *Proc. IEEE CICC*, pp. 297-300, May 1994.
- [42] E. Charbon, E. Malvasi and A. Sangiovanni-Vincentelli, "Generalized Constraint Generation for Analog Circuit Design", in *Proc. IEEE ICCAD*, pp. 408-414, November 1993.
- [43] B. Basaran, R. A. Rutenbar and L. R. Carley, "Latchup-Aware Placement and Parasitic-Bounded Routing of Custom Analog Cells", in *Proc. IEEE ICCAD*, pp. 415-421, November 1993.
- [44] Z. Daoud and C. J. Spanos, "DORIC: Design of Optimal and Robust Integrated Circuits", in *Proc. IEEE CICC*, pp. 361-364, May 1994.
- [45] E. H. L. Aarts and P. J. M. van Laarhoven, *Simulated Annealing: Theory and Applications*, D. Reidel Publishing, 1987.
- [46] C. Sechen and A. Sangiovanni-Vincentelli, "Chip-Planning, Placement and Global Routing of Macro/Custom Cell IC's using Simulated Annealing", in *Proc. IEEE/ACM DAC*, pp. 73-80, June 1988.
- [47] M. Burstein, "Channel Routing", in *Layout Design and Verification*, ch. 4, pp. 133-167, T. Ohtsuki Ed., North Holland, 1986.
- [48] G. W. Clow, "A Global Routing Algorithm for General Cells", in *Proc. IEEE/ACM DAC*, pp. 45-51, 1984.
- [49] E. Felt, E. Charbon, E. Malvasi and A. Sangiovanni-Vincentelli, "An Efficient Methodology for Symbolic Compaction of Analog IC's with Multiple Symmetry Constraints", in *Proc. European Design Automation Conference*, pp. 148-153, September 1992.
- [50] E. Malvasi, E. Felt, E. Charbon and A. Sangiovanni-Vincentelli, "Symbolic Compaction with Analog Constraints", *International Journal of Circuit Theory and Applications*, John Wiley & Sons, vol. 23, n. 4, pp. 433-452, July-August 1995.
- [51] J. Kuhn, "Analog Module Generators for Silicon Compilation", *VLSI Systems Design*, pp. 74-80, May 1987.
- [52] U. Choudhury and A. Sangiovanni-Vincentelli, "An Analytical-Model Generator for Interconnect Capacitances", in *Proc. IEEE CICC*, pp. 861-864, May 1991.

**Enrico Malvasi** received the "Laurea" degree (summa cum laude) in Electrical Engineering from the University of Bologna, Italy in 1984. In 1993, he received the M.S. degree in Electrical Engineering from the University of California at Berkeley. Between 1986 and 1989 he was with the Dept. of Electrical Engineering and Comp. Science (DEIS) of the University of Bologna, working on computer-aided-design for analog circuits. In 1989, he joined the *Dipartimento di Elettronica ed Informatica* of the University of Padova, Italy, as Assistant Professor. Since 1990 he has collaborated with the CAD group of the Dept. of EECS of the University of California at Berkeley, where he carried out research on performance-driven CAD tools and methodologies for analog design. In 1995 he joined Cadence Design Systems Inc., San Jose, Ca, as architect for physical design automation.

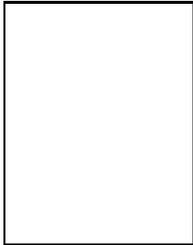
His research interests include several areas of analog design automation: layout, design methodologies, optimization and circuit analysis.

**Edoardo Charbon** obtained the *Diploma* in Electrical Engineering from the Swiss Federal Institute of Technology (ETH) at Zurich, Switzerland in 1988. Between 1988 and 1989 he worked at the Dept. of Electrical Engineering of the ETH, where he designed CMOS A/D converters for integrated sensor applications. In 1989 he visited the Dept. of Electrical Engineering of the University of Waterloo, Canada where he was involved in the design and fabrication of ultra low-noise, nano-Tesla magnetic

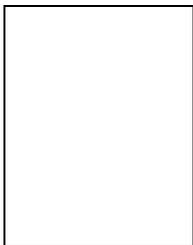
sensors. In 1991 he obtained the M.S. degree in Electrical and Computer Engineering from the University of California at San Diego and in the same year he joined the Dept. of Electrical Engineering and Computer Sciences at the University of California, Berkeley where he earned the Ph.D. degree in 1995.

His research interests include CAD for analog and mixed-signal ICs, RF design, microwave and superconducting parasitic analysis and micromachined sensor design.

Technical Program Chairperson of the International Conference on CAD and its General Chair. He has been on the program committee of several conferences and has been on the editorial board of the IEEE Transactions on CAS, the IEEE Transactions on CAD and of several other archival Journals.



**Eric Felt** graduated from Duke University in 1991 with a B.S.E. in electrical engineering, Summa Cum Laude, with Distinction. He received his M.S. degree in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 1993 and is currently working towards his Ph.D. degree in the same department. His current research topic is CAD for the testing and characterization of analog IC's.



**Alberto Sangiovanni-Vincentelli** is a Professor of Electrical Engineering and Computer Sciences at the University of California at Berkeley where he has been on the Faculty since 1976. He obtained a "Dottore in Ingegneria" electrical engineering and computer science degree *summa cum laude* from the Politecnico di Milano, Milano, Italy in 1971. He was Assistant Professor from 1971 to 1974 and a "Professore Incaricato" from 1974 to 1976 at the Politecnico di Milano. In 1980-1981, he spent a year as a Visiting Scientist at the Mathematical Sciences Department of the IBM T.J. Watson Research Center. In 1987, he spent six months at MIT as Visiting Professor. He has held a number of visiting professor positions at the University of Torino, University of Bologna, University of Pavia, University of Pisa and University of Rome.

He helped founding Cadence and Synopsys, the two leading companies in the area of Electronic Design Automation. He was a Director of ViewLogic and Pie Design System and Chair of the Technical Advisory Board of Synopsys. He is on the Board of Directors of Cadence a founder and a member of the Management Board of the Cadence Berkeley Laboratories. He has consulted for a number of US companies including IBM, Intel, ATT, GTE, GE, Harris, Nynex, Teknekron, DEC, HP, Japanese companies including Kawasaki Steel and Fujitsu, and European companies including SGS-Thomson, Alcatel, Magneti-Marelli, ENI, Montedison, FIAT, Bull.

He is on the Advisory Board of the Lester Center of the Haas School of Business, of the Center for Western European Studies and a member of BRIE (Berkeley Roundtable for the International Economy), all of the University of California.

In 1981 he received the Distinguished Teaching Award of the University of California. He received the 1995 Graduate Teaching Award of the IEEE for "inspirational teaching of graduate students". He has received three Best Paper Awards (1982, 1983 and 1990) and a Best Presentation Award (1982) at the Design Automation Conference and is the co-recipient of the Guillemin-Cauer Award (1982-1983), the Darlington Award (1987-1988) and the Best Paper Award of the Circuits and Systems Society of the IEEE (1989-1990).

His research interests are related to the application of mathematical ideas and approaches to the design of electronic systems. In particular, he focused on numerical analysis for the solution of large scale systems of non linear equations arising from the analysis of large scale Integrated Circuits, the design and optimization of digital circuits, new approaches to the problem of integrated circuits layout, and to the parallelization of complex algorithms in computer-aided design. Recently, he has been focusing on design methodologies and tools for mixed signal integrated circuits including high-frequency and low power circuits, and for embedded controllers. In this latest area, he is interested in methods for software synthesis and formal verification.

He has published over 350 papers and five books in the area of design methodologies and tools. Dr. Sangiovanni-Vincentelli was the