Generalized Constraint Generation for Analog Circuit Design

Edoardo Charbon, Enrico Malavasi† and Alberto Sangiovanni-Vincentelli
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
†Dip. di Elettronica e Informatica, Università di Padova, Italy

Abstract

A general methodology is presented for the generation of a complete set of constraints on interconnect parasitics, parasitic mismatch and on the physical topology of analog circuits. The parasitic and matching constraints are derived from high-level performance specifications by means of sensitivity analysis in time and frequency domain using quadratic optimization. Topological constraints are obtained by using sensitivity and matching information on devices and interconnect as well as graph-based techniques to extract the necessary geometric information.

1 Introduction

The design of analog circuits is often a difficult task compared with a digital one of similar complexity because of the higher number of specifications and the importance of second order effects. In addition, the continuously growing complexity of analog integrated circuits has required a better control over the design quality and the redefinition of tasks like module generation and floorplanning. The performances of analog circuits can be extremely sensitive to parasitic effects. In particular, due to continuous reduction of the size of circuit devices, layout induced parasitics increasingly influence the circuit behavior.

The performance degradation of a circuit due to layout parasitics can be modeled using *performance sensitivities* [1]. Sensitivities can also be used to generate a set of constraints on interconnect parasitics [2]. Techniques [3, 4, 5, 6, 7] have been proposed for a constraint-based approach to the layout of analog integrated circuits. In these approaches a constraint generator is used to map a set of high-level performance specifications onto a set of bounds, which are then used during the layout synthesis to control parasitics.

The constraint generation problem has been first formally stated in [2]. In this formulation, performance sensitivities are used to map directly performance specifications onto a set of constraints on critical parasitics.

Although important, interconnect induced parasitics are not the only source of degradation of circuit performance. Another class of parasitics, namely those induced by the topology of the circuit, is a relevant, and often dominant source of non-ideal circuit behavior. Generally, constraints on the layout topology are imposed by the designer based on experience. However in complex designs, such as mixed-signal circuits, this task may become difficult if not impossible, due to the interactions between circuit components and parasitics. During the synthesis of complex mixed-signal systems [8] it is often required to map high-level performance specifications onto several distinct sets of layout constraints. The constraint generation process should therefore be flexible.

In this paper a generalization of the constraint generation problem for the layout synthesis of analog integrated circuits is presented. A unified definition strategy is also defined to express and manipulate performance specifications, so as to maximize the flexibility of the constraint-based tools used during the synthesis phase.

The set of layout constraints, in its most general form, includes two types of constraints: topological and parasitic constraints. Topological constraints relate to the *relative position* and *orientation* of the circuit components. Parasitic constraints are referred to as a bound to the *value* or to the *mismatch* of parasitic components. We show how to manipulate performance sensitivities to model the performance degradation due to parasitic mismatch. The sensitivity information, along with estimates on the minimum and maximum possible parasitics, is then used to determine critical parasitic mismatches.

Topological constraints are generated by modeling device mismatch as a parasitic mismatch, thus allowing us to identify all critical matching constraints in the circuit. Symmetry constraints, a subset of all matching constraints, are identified by mapping the circuit onto a weighted undirected graph. The nodes of the graph represent the circuit devices, the edges the connectivity and the degree of matching. The sources of the graph are the inputs, the virtual and real grounds being the sinks. The symmetry constraints are obtained by traversing the graph on two paths simultaneously from sources to sinks, the search direction being determined by device matching considerations and connectivity.

The flow diagram of the constraint generation process is shown in figure 1. From hardware and performance description a sensitivity analysis of the circuit is performed using mixed symbolic / numerical techniques [9, 10]. Using performance sensitivity information and *a priori* estimates of minimum and maximum parasitic values, the subset of critical parasitics is detected. A quadratic programming approach is then used to compute bounds on critical parasitics and to locate matching constraints, so that the total performance degradation collectively induced by all interconnect parasitics does not exceed the specifications. The circuit is then mapped onto a weighted graph and solved to obtain all symmetry constraints. Finally, all constraints undergo a technology mapping step to compute detailed geometrical bounds on all layout components. Thus all specifications on the yield properties of the final circuit can be satisfied.

The paper is structured as follows. The foundations of the constraint generation problem and the principles of the computation of performance sensitivities are reviewed in section 2. The algorithms for the generation of parasitic and topological constraints are outlined in sections 3 and 4. Benchmark ex-

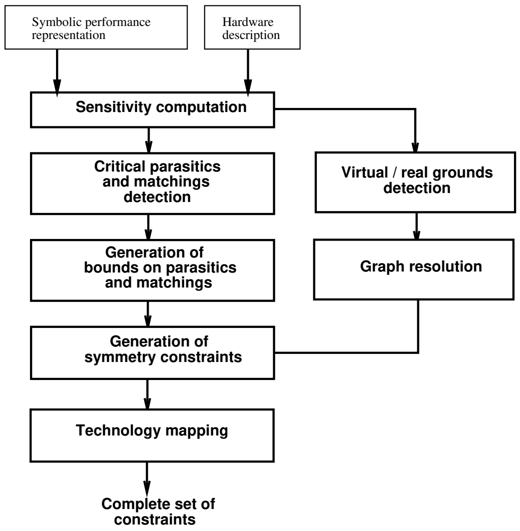

Figure 1: Flow diagram of the constraint generation process.

amples showing the suitability of our approach are discussed in section 5.

2 Constraint Generation

2.1 General Problem Formulation

For a set of performances $\{W_i\}$, $i = 1, ..., N_w$ and a set of parasitics $\{p_j\}$, $j = 1, ..., N_p$, *parasitic constraint generation* is defined as the process of creating a suitable set of constraints on $\{p_j\}$:

$$
\begin{aligned}
& p_j \le p_j^{(bound)}, \\
& p_k = p_l, \; k \neq l.
\end{aligned} \tag{1}
$$

To guarantee the fulfillment of performance constraints, equation (2) must be satisfied

$$\triangle W_i \le \overline{\triangle W_i} \, , \qquad \forall \, i. \tag{2}$$

$\triangle W_i$ represents the total and $\overline{\triangle W_i}$ the maximum allowed degradation of performance W_i with respect to all parasitics and mismatches.

In this formulation all the parameters which can be influenced during the layout synthesis are considered. Interconnect parasitics, cross-coupling induced by interconnections, parasitic as well as device mismatches in any of their parameters must ultimately satisfy equation (1).

2.2 Use of Sensitivities to Model Performance Constraints

Assuming that performance functions are close to their nominal values, then linear approximations of their dependence on parasitics can be obtained using sensitivities.

Let us consider a parasitic element p_j, with nominal zero value. The sensitivity of performance W_i with respect to p_j is defined as

$$S_j^i = \left[\frac{\partial W_i}{\partial p_j} \right]_{p_j = 0} . \tag{3}$$

Define $S_j^{i\,+} = S_j^i$ for positive values of the sensitivity. Alternatively $S_j^{i\,-} = -S_j^i$ for the negative ones. The positive and negative components of the performance degradation $\triangle W_i^+$ and $\triangle W_i^-$ with respect to all parasitics belonging to the set $\{p_j\}$ are, in first approximation

$$\triangle \, W_i^+ = \sum_j S_j^{i\,+} p_j \, , \quad \triangle W_i^- = \sum_j S_j^{i\,-} p_j \tag{4}$$

The distinction between positive and negative components of the degradation must be considered throughout the performance generation process to obtain constraints which reflect a worst-case analysis. For simplicity of notation the sign is dropped hereafter, although it is implicitly assumed that all operations are carried out. In addition, technology-related process variations can induce deviations of the performance functions. These effects can be taken into account by simply replacing nominal values of sensitivities with worst-case values as suggested in [2]. The parasitic constraint generation is based on this approximation as shown in section 3.

Let us now consider the degradation of performance W_i due to parasitic mismatch. Let p_1, p_2 be two matched parasitics. Although they are nominally equal, they actually differ by a small mismatch $\triangle p$. We introduce the following notation:

$$
\left\{
\begin{array}{l}
p = (p_1 + p_2)/2 \\
\triangle p = (p_1 - p_2)
\end{array}
\right. \tag{5}
$$

so that p_1 and p_2 can be expressed in terms of their average value p and their mismatch $\triangle p$ as: $p_1 = p + \frac{\triangle p}{2}$, $p_2 = p - \frac{\triangle p}{2}$. The sensitivities of performance W_i with respect to p and $\triangle p$ are respectively

$$
\left\{
\begin{array}{l}
S_p^i = \dfrac{\partial W_i}{\partial p} = \dfrac{\partial W_i}{\partial p_1}\dfrac{\partial p_1}{\partial p} + \dfrac{\partial W_i}{\partial p_2}\dfrac{\partial p_2}{\partial p} = S_1^i + S_2^i \\
S_\triangle^i = \dfrac{\partial W_i}{\partial \triangle p} = \dfrac{\partial W_i}{\partial p_1}\dfrac{\partial p_1}{\partial \triangle p} + \dfrac{\partial W_i}{\partial p_2}\dfrac{\partial p_2}{\partial \triangle p} = \dfrac{S_1^i - S_2^i}{2}
\end{array}
\right.
$$

When matching constraints involve simultaneously more than two parasitics, a more general definition of the average value and mismatch (5) is adopted. For a group of parasitics $p_1, p_2, \cdots, p_n$, their average value p and their mutual mismatches $\triangle p_{jk}$ are defined as follows:

$$
\left\{
\begin{array}{l}
p = \dfrac{1}{n} \displaystyle\sum_{j=1}^{n} p_j \\
\triangle p_{jk} = p_j - p_k \qquad j, k = 1, \cdots, n.
\end{array}
\right. \tag{6}
$$

Consequently, the sensitivities of performance W_i with respect to p and $\triangle p_{jk}$ are

$$
\left\{
\begin{array}{lll}
S_p^i & = \dfrac{\partial W_i}{\partial p} = \displaystyle\sum_{j=1}^{n} \left(\dfrac{\partial W_i}{\partial p_j}\dfrac{\partial p_j}{\partial p} \right) = \displaystyle\sum_{j=1}^{n} S_j^i \\
S_{\triangle jk}^i & = \dfrac{\partial W_i}{\partial \triangle p_{jk}} = \dfrac{(S_j^i - S_k^i)}{2}
\end{array}
\right. \tag{7}
$$

Given two parasitics p_1 and p_2, their contribution $\triangle W_i$ to the degradation of performance W_i is, in first approximation

$$\triangle W_i = S_1^i p_1 + S_2^i p_2 .$$

Consequently,

$$\triangle \, W_i = S_p^i (p_1 + p_2) + \frac{S_\triangle^i}{2}(p_1 - p_2) = 2 S_p^i p + \frac{S_\triangle^i}{2} \triangle \, p. \tag{8}$$

It is evident that if

$$S_p^i \ll S_\triangle^i , \tag{9}$$

the contribution of p_1, p_2 to the degradation of W_i can be significantly reduced by increasing the correlation between the two parasitics, i.e. by enforcing the matching between them. Therefore matching constraints can be automatically determined on the ground of the relative importance of sensitivity on the average value and sensitivity on mismatch in the expression of performance degradation (8). The inequality (9) can be used to distinguish between the cases when mismatches yield significant degradations and the cases when parasitic correlation has little or no influence on the performance whatsoever.

Matching constraints are often expressed as a difference between parameter ratios rather than of simple parameters. This is often the case when trying to establish specific geometric constraints from topological constraints during the technology mapping process. Given two independent variables x and y and a performance $W(x/y)$ then

$$S^W_{(x/y)} = \frac{\partial W}{\partial (x/y)} = \frac{y^2}{y-x}(S^W_x + S^W_y), \tag{10}$$

and thus, the sensitivity of performance W with respect to the mismatch $\triangle \frac{x}{y} = \frac{x_1}{y_1} - \frac{x_2}{y_2}$ can be written as

$$S^W_{\triangle \frac{x}{y}} = (\frac{S^W_{x_1/y_1} - S^W_{x_2/y_2}}{2}) = \frac{y_1^2}{2\ (y_1 - x_1)}(S^W_{x_1} + S^W_{y_1})$$

$$- \frac{y_2^2}{2\ (y_2 - x_2)}(S^W_{x_2} + S^W_{y_2}). \tag{11}$$

Throughout this analysis it was assumed that x_i and y_i are independent variables. This is usually the case in layout since orthogonal geometries, such as channel width and length in MOS transistors, are generally influenced by independent sources.

2.3 Canonical Representation of Performance

A generalized expression for the computation of sensitivities from a set of arbitrary performances has been derived in [11, 12]. This formulation has been used by us to represent all performances analyzed in a compact and rigorous way, thus ensuring flexibility of our design tools. For completeness the formulation has been reviewed hereafter.

Let us consider an arbitrary performance W, let $\mathbf{x}$ be a vector of design parameters (e.g. capacitances, MOS channel length, etc.), $\mathbf{y}$ a vector of circuit variables (e.g. voltages, charges, etc.), and ω an independent variable (e.g. frequency, time). Then the equation

$$g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \omega), \omega) \tag{12}$$

fully describes the performance W. Define now the characteristic function of performance W as

$$h(\mathbf{x}, \mathbf{y}(\mathbf{x}, \omega), \omega) = 0 \tag{13}$$

The characteristic function h represents implicitly the operating point at which the sensitivity computations have to be performed.

The performance sensitivity of the performance W with respect to a set of design parameters $\mathbf{x}$ is

$$S^W_{\mathbf{x}} = \frac{\partial g}{\partial \mathbf{x}} + \frac{\partial \mathbf{y}}{\partial \mathbf{x}}^T \frac{\partial g}{\partial \mathbf{y}} - \left\{ \frac{\frac{\partial g}{\partial \omega} + \frac{\partial g}{\partial \mathbf{y}}^T \frac{\partial \mathbf{y}}{\partial \omega}}{\frac{\partial h}{\partial \omega} + \frac{\partial h}{\partial \mathbf{y}}^T \frac{\partial \mathbf{y}}{\partial \omega}} \right\} \left[\frac{\partial h}{\partial \mathbf{x}} + \frac{\partial \mathbf{y}}{\partial \mathbf{x}}^T \frac{\partial h}{\partial \mathbf{x}} \right] \tag{14}$$

The expression $\frac{\partial f}{\partial \mathbf{y}}$ has the same meaning of $\nabla f(\mathbf{y})$, if $f:\ \mathbf{R}^n \rightarrow \mathbf{R}$.

3 Generating Parasitic Constraints

3.1 Bounding Constraints

At an early stage of the design, generally no precise information is available on the exact values of the parasitics. However very reasonable assumptions can be made on their range. A bound on the value of a parasitic, or bounding constraint should be therefore within this range, thus ensuring realizability of the circuit. In symbols

$$p_j^{(min)} \leq p_j^{(bound)} \leq p_j^{(max)}, \quad \forall j. \tag{15}$$

$p_j^{(min)}$ and $p_j^{(max)}$ are determined by technology considerations and $p_j^{(bound)}$ is the bounding constraint to be computed. From this information, using expression (4), one can obtain the minimum and maximum performance degradation induced by all parasitics.

Furthermore, we discard those parasitics whose cumulative effect on performance degradation is negligible, compared to the maximum allowed performance degradation [1]. For the remaining critical parasitics, bounds can be automatically generated as a result of a constrained optimization, the objective being a model of the tool flexibility.

3.2 Matching Constraints

The general methodology to extract matching constraints can be defined as follows. Let M be a *matching cluster*, i.e. a group of circuit components constrained by matching requirements. Performance sensitivities with respect to parasitic mismatch $\triangle p_{ij} = p_i - p_j$ and average $p^{(M)} = \frac{1}{n}\sum_{j \in M} p_j$, are computed using the techniques outlined in section 2.2. The range of variation of the average and mismatch is computed as

$$\frac{1}{n}\sum_{j \in M} p_j^{(min)} \leq p^{(M)} \leq \frac{1}{n}\sum_{j \in M} p_j^{(max)},$$

$$p_i^{(min)} - p_j^{(max)} \leq \triangle p_{ij}^{(mismatch)} \leq p_i^{(max)} - p_j^{(min)}, \quad \forall j. \tag{16}$$

One can recognize that the constraint generation problem for the parasitic average and for a simple parasitic component are identical. The constraint generation for parasitic mismatch on the other hand is handled in a slightly different manner. Suppose a solution for $\triangle p_{ij}$ has a negative value, then this term could possibly generate cancellation in the approximation formula for performance degradation $\triangle W_i$ in (4), thus creating false results. To eliminate this problem, the variable $\triangle p_{ij}$ is split into two, $\triangle p_{ij}{}^+$ and $\triangle p_{ij}{}^-$, which are consequently added as a contribution to the approximate negative and positive components of the performance degradation. $\triangle p_{ij}{}^-$ and $\triangle p_{ij}{}^+$ represent the lower- and upper-bound of the allowed mismatch.

This approach is often expensive computationally, since it involves the generation of a large number of mismatches, and the constraint generation problem has to be solved on a large set of parameters.

A more efficient approach is to use the matching requirement expressed in (9). For every couple of parasitics, their mismatch and average sensitivities are computed and compared against each other. If relation (9) holds, the original parasitics are discarded and substituted by their average value and the mismatch. Otherwise they are kept and mismatches between them are not considered. With this approach, the computational cost of the constraint generation problem is affected slightly, only one being the parameter added for each group of three or more matched parasitics.

4 Generating Topological Constraints

4.1 Device Matching

The importance of device matching in integrated circuits has been shown not only for active but also for passive elements [13, 14]. Designers generally impose matching constraints on circuit devices to ensure that voltage and current mismatches in these devices be bounded. These constraints can be mapped directly onto constraints on the geometry of the physical implementations [14]. For this reason device matching constraints are usually of qualitative nature, mostly dictated by the expertise of the designer.

We propose a quantitative approach to the generation of device matching constraints based on the principles introduced in section 2.2. Assume for simplicity but without loss of generality that all active devices can be represented by a two-port. Furthermore, assume that a model relating the output to the input port is available and that only one performance W is considered. Let V_{D_i} and I_{D_i} be the quantities characterizing respectively input and output of device D_i. Let $I_{D_i} = f(\mathbf{P_0} + \mathbf{P}, V_{D_I})$, where $\mathbf{P}$ is the vector of all deviations from a nominal value $\mathbf{P_0}$, of all technological parameters affecting the device. Suppose the set of sensitivities $\{S_{P_{ik}}\}$ of performance W with respect to all vector elements of $\mathbf{P}$ is available. Then, in first approximation, the degradation $\triangle W_{D_i}$ of performance W with respect to the technological deviations for device D_i can be expressed as

$$\triangle W_{D_i} = \sum_k S_{P_{ik}} \; P_k. \tag{17}$$

For reasons that will be clear later (equation 19), the sign of sensitivities has been dropped.

Consider now a pair of devices D_i and D_j, then the degradation due to the parameter mismatch of the devices can be computed as

$$\triangle W_{D_{ij}} = \sum_k S_{\triangle P_k} \; \triangle P_k. \tag{18}$$

The $S_{\triangle P_k}$ are computed using (7) and $\triangle P_k$ is the k^{th} component of the vector difference $\mathbf{P(D_i)} - \mathbf{P(D_j)}$.

Assume that the $\triangle P_k$'s are independent random variables with zero mean. The variance of the degradation of performance W with respect to the variances of the mismatches of all technological parameters relevant to the pair of devices D_i and D_j, is computed as

$$\sigma^2(\triangle W_{D_{ij}}) = \sum_k |S_{\triangle P_k}|^2 \; \sigma^2(\triangle P_k). \tag{19}$$

Consider for instance a pair of matched MOS transistors. The variance of the degradation due to technological mismatches can be expressed as

$$\sigma^2(\triangle W_{m_{12}}) = S_{\triangle W}{}^2\sigma^2(\triangle W) + S_{\triangle L}{}^2\sigma^2(\triangle L) +$$
$$+ S_{\triangle C_{ox}}{}^2\sigma^2(\triangle C_{ox}) + S_{\triangle \mu_n}{}^2\sigma^2(\triangle \mu_n) + S_{\triangle V_{TO}}{}^2\sigma^2(\triangle V_{TO}).$$

W, L, C_{ox}, μ_n and V_{TO} are respectively channel width, channel length, gate oxide capacitance, mobility and threshold voltage of the transistors.

In [14] a direct relation has been shown between these variances and the relative orientation and distance between device pairs. This information can be used to translate the maximum allowed performance degradation into the physical separation and relative orientation between pairs of devices. In order to do this, estimates of the minimum and maximum attainable variances for a particular process are needed, so that an upper- and lower-bound of the performance degradation for **each** pair of devices can be determined. The sum of all degradations due to each pair of devices in the circuit should be added to equation (4). The constrained optimization is solved to find the actual variances of each parameter. Consequently numerical values for distances and orientations.

Notice that, for consistency, the standard deviation should be added to (4), thus making the optimization harder and more time consuming. In order to avoid this additional complexity, the expression for the standard deviation is linearized after substituting the single parameter variances with analytical models in the geometric quantities of interest. Thus a linear expression for the standard deviation of the degradation referred to the pair is obtained as

$$\sqrt{\sigma^2(\triangle W_{m_{ij}})} \simeq A_{ij} d_{ij} + B_{ij} r_{ij}. \tag{20}$$

A and B are quantities which depend upon the performance sensitivity of W with respect to the pair's technology parameters. d_{ij} and r_{ij} represent respectively the distance and relative rotation of devices m_i and m_j . Also in this case a simplification mechanism similar to the one proposed in [2] is used for computing the criticality of the mimatches.

4.2 Device and Interconnect Symmetry

Symmetry is used in the layout of analog integrated circuits to minimize the effects of mismatched parasitics on circuit performance. With symmetric placement and routing differential signal paths can be matched optimally. The detection of topological symmetry constraints in an automatic fashion has been traditionally associated with pattern recognition or expert system based techniques.

We propose a novel method based on our quantitative approach for the computation of parasitic and device matching constraints combined with information on the electrical properties of the circuit nets. The goal of topological symmetry is to facilitate the respect of matching constraints on parasitics and / or devices. Hence priority to undergo a symmetrization process should be given to those *entities* (devices or interconnect) on which *matching constraints* have been imposed. A second requirement is that these entities belong to *distinct differential signal paths* in order to maximally balance the signals circulating in the circuit. A third requirement is that *symmetric signal paths ultimately converge to a virtual or real signal ground*. These three basic requirements have been used to drive a graph-based search algorithm described as follows

```
1. map_onto_graph;
2. add_edges;
3. create_sink_nodes;
4. set_starting_nodes;
5. until (all_nodes_visited) {
     if (termination_condition) {
        find_new_starting_node;
        continue;
     } else {
        search_matched_nodes;
        run_through_reconvergent_path;
        if (not_degenerate) continue;
        if (semi-degenerate &&
            super_virtual) continue;
        invalidate_path;
     }
   }
6. exit
```

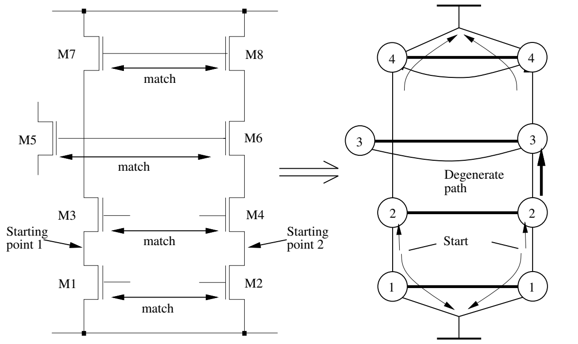

Figure 2: Degenerate path.

In the first step the circuit description is converted into an undirected graph, whose nodes represent the (active and passive) device and edges connectivity. In the second step the matching constraints are introduced into the graph by adding constraint edges to it. In the third step edges associated with real and virtual grounds are found and are set to be the graph sinks. This operation is automatically performed in all nets by comparing common and differential mode gains with respect to the input.

After initializing the search, all nodes are searched until the maximum number of symmetric paths has been found, the termination condition for each path being the presence of a virtual or a real ground. In the first case the symmetry path is called *reconvergent*. In order to be completely satisfied, the symmetric search paths must merge and proceed on the graph until a real ground is found. In the second case the symmetry constraints for the current search path(s) are completely satisfied. Thus the search ends and a new one begins from another couple of non-terminal nodes satisfying matching constraints.

If a *degenerate* path is detected the search path is invalidated, thus nulling the symmetry constraints found associated with it. A degenerate configuration occurs whenever a device on a search path has other devices matched to it on an incompatible search path (M5 and M6 in figure 2). Thus separate symmetry constraints interfere with each other.

If a *semi-degenerate* path is detected, i.e. a node is found to be at both sides of the symmetry axis (M1 and M2 in figure 3), a pair of edges associated with a *super-virtual ground* is searched. The symmetry constraint associated with the current search path is invalidated when no such edges are found. A *super-virtual ground* is defined as a pair of signal nets such that, if connected by a series of two identical resistors, the middle point becomes a virtual ground.

The final steps are used to map all search paths onto actual topological symmetry constraints.
Figure 2 shows a simple circuit and the graph associated with it, when a degenerate path exists. Nodes "1" are visited first. Since a terminal condition occurs (real ground), the search continues in the opposite direction. After one iteration at nodes "2", the search proceeds towards the nodes "3", however since neither nodes are in a contiguous path a degenerate situation occurs. The last section of the path is therefore invalidated and the search continues from nodes "4", ending at the top, when a terminal condition occurs. Thus the search is complete resulting in the symmetry constraints M1-M2; M3-M4 and M7-M8.

Figure 3 illustrates a semi-degenerate situation. In this case electrical nets A and B clearly form a super-virtual ground. Thus, the constraint cycle in the symmetry constraint dependence is eliminated and the symmetry constraint is accepted.

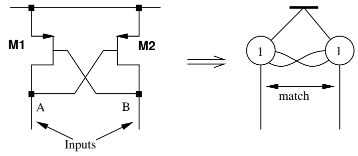

Figure 3: Semi-degenerate path.

5 Results

The algorithms outlined in this paper have been tested on a wide range of circuits, selected from a set of commonly used industrial applications. Because of limited space, we will discuss in detail two particularly significant circuits to highlight the features of the proposed algorithms.

For the computation of voltage sensitivities, standard analysis methods available in SPICE3 [15] have been used. Performance sensitivities are computed from circuit variable sensitivities using the principles presented in section 2.3. The sensitivity information required for matching and symmetry constraints computations is obtained from performance sensitivities using the manipulations outlined in section 2.2. Presently available are dc, ac and transient analysis. The set of design parameters currently supported are *resistances, substrate capacitances, capacitive coupling, inductances, mutual inductances, base-emitter junction area, MOS channel length* and *width*.

Class AB amplifier

Consider the power amplifier "ab", depicted in figure 4. The

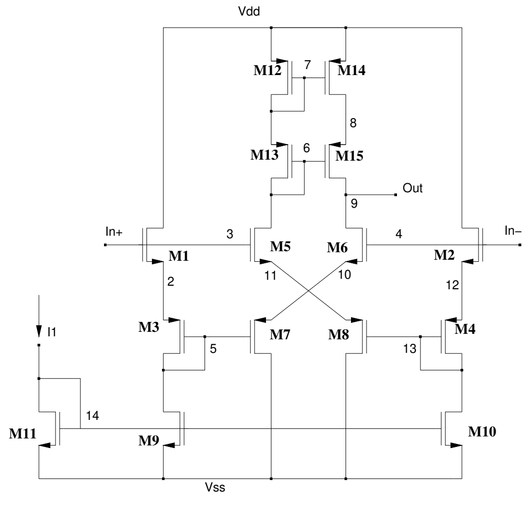

Figure 4: Class AB amplifier "ab".

performances analyzed are offset voltage, unity gain bandwidth, phase margin and low frequency gain. Table 1 reports nominal values, positive and negative constraints on the degradation of each performance. Table 1 shows some of the most critical parasitics and the bounds on the mismatch for capacitive and resistive parasitics. The CPU time for the computation of all constraints was 7.4 seconds on a DECstation 5000/125. Notice that parasitic capacitances are defined by the nets they are connected to. Resistances on the other hand, refer to the branches in which they are introduced. Table 2 illustrates each net type as evaluated by the program. Notice in particular the evaluation of nets 10 and 11, correctly assigned to the type "virtual ground". In this case the knowledge of the nature

performance constraints			
Performance	Nominal value	$\triangle W^+$	$\triangle W^-$
Offset Voltage	14 mV	-2 mV	8 mV
U.G. bandwidth	500 MHz	-3 MHz	3 MHz
Phase margin	106 deg	-3deg	3deg
L.F. gain	28 dB	-3 dB	∞
capacitive mismatch			
Net pair 1	Net pair 2	bound	limits
3 - 9	4 - 6	62.41fF	± 100fF
11 - gnd	10 - gnd	80.97fF	± 100fF
resistive mismatch			
branch 1	branch 2	bound	limits
Vdd-m12.s	Vdd-m14.s	.100Ω	± 10Ω
Vss-m9.s	Vss-m10.s	.100Ω	± 10Ω
10-m7.s	11-m8.s	4.201Ω	± 10Ω
10-m6.s	11-m5.s	4.201Ω	± 10Ω
12-m4.s	2-m3.s	25.563Ω	± 50Ω
12-m2.s	2-m1.s	25.563Ω	± 50Ω

Table 1: "ab": Bounds on the resistive and capacitive mismatch. Note: ".s" indicates the source of a transistor.

net	0	2	3	4	5	6	7	8
type	G	S	I	I	S	S	S	S

net	9	10	11	12	13	14	Vdd	Vss
type	O	V	V	S	S	G	G	G

Table 2: "ab": Automatic net type assignment (S = Signal, G = Supply/Gnd, V = Virtual Gnd, I = Input, O = Output).

Device matching constraints		
distance	rel. rotation	devices
200 λ	no	m12 - m14
200 λ	no	m9 - m10
200 λ	no	m5 - m6
200 λ	no	m7 - m8
200 λ	yes	m1 - m2
200 λ	yes	m3 - m4

Symmetry constraints	
axis	Symmetric devices
0	m1 - m2; m3 - m4; m7 - m8; m9 - m10
1	m12 - m14; m13 - m15; m5 - m6

Table 3: "ab": Topological constraints.

of these nets was crucial to correctly recognize symmetry constraints, otherwise difficult to individuate due to the irregular topology of the circuit. In conclusion, table 3 shows a summary of the topological constraints obtained using the techniques described throughout the paper. For each matching constraint the corresponding values for d and r are given based on the a standard scmos $2\mu m$ technology. The CPU time for the topological symmetry computation was 0.1 seconds on a DECstation 5000/125. Notice the distinct symmetries found by our program, one for each sub-circuit of the amplifier. Generally the axes of these symmetries are superimposed to achieve more compact layouts.

Clocked comparator

Consider now the comparator "fastcomp", depicted in figure 5. This circuit is of particular interest because of

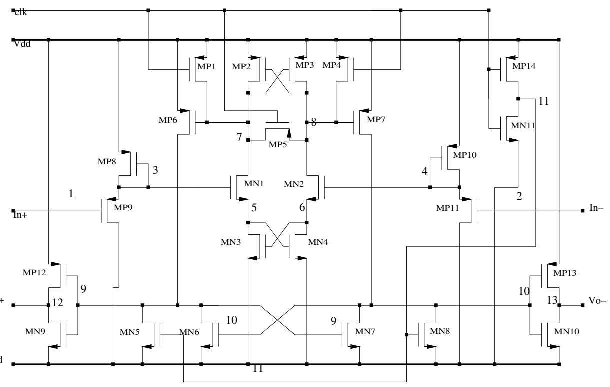

Figure 5: Clocked comparator "fastcomp".

the presence of several irregular structures creating semi-degenerate paths. Another point of interest is the fact that the analog circuit is switched. Thus the topological generation mechanism is to be tested in a circuit which operates in a strongly non-linear region. In addition, there are no virtual grounds in the circuit, thus making the symmetry constraint generation more difficult. Table 4 lists the performances of interest for this circuit and the used constraints. The calculated mismatches are shown in table 4. The CPU time required was 33.1 seconds on a DECstation 5000/125. Notice that major contribution to the offset is given by parasitic resistances degenerating the input stage devices. As expected, the input source followers (MP10-11 and MP8-9) are less critical than the high-gain pairs (MN3-4 and MP2-3). Table 6 shows the results of the topological constraint generation. The required CPU time was 0.5 seconds on a DECstation 5000/125. Notice how all semi-degenerate configurations have been resolved, yielding three groups of symmetric devices, which have to be placed necessarily on the same axis in order to minimize the offset.

6 Conclusion and Future Work

A generalized approach to the automatic generation of parasitic and topological constraints for designing analog integrated circuits has been presented. The method uses a symbolic representation of the performance as sufficient information for the computation of numerical values for the sensitivities, which are then used for the generation of bounding and matching constraints. Topological constraints are computed using the matching information on devices as well as graph-based techniques for identifying symmetries.

Constraint-based layout synthesis tools can exploit this information to drive the layout optimization to a config-

performance constraints			
Performance	Nominal value	$\triangle W^+$	$\triangle W^-$
Offset Voltage	20mV	-3 mV	3 mV
Delay (LH)	2.58ns	-	1 ns
Delay (HL)	2.92ns	-	1 ns
capacitive mismatch			
Net pair 1	Net pair 2	bound	limits
7 - gnd	8 - gnd	27.15fF	± 100fF
7 - vdd	8 - vdd	27.15fF	± 100fF
1 - 5	2 - 6	68.32fF	± 100fF
resistive mismatch			
branch 1	branch 2	bound	limits
MN3.s-gnd	MN4.s-gnd	.100Ω	± 1Ω
MP2.s-vdd	MP3.s-vdd	7.708Ω	± 50Ω
MP8.s-Vdd	MP10.s-Vdd	42.90Ω	± 50Ω

Table 4: "fastcomp": Bounds on capacitive and resistive mismatch. Note: ".s" indicates the source of a transistor.

net	0	1	2	3	4	5	6	7	8
type	G	I	I	S	S	S	S	S	S
net	9	10	11	12	13	clk	Vdd	Gnd	
type	S	S	S	O	O	S	G	G	

Table 5: "fastcomp": Automatic net type assignment (S = Signal, G = Supply/Gnd, V = Virtual Gnd, I = Input, O = Output).

Device matching constraints		
distance	rel. rotation	devices
100 λ	no	MN3 MN4
100 λ	no	MP2 MP3
100 λ	yes	MP8 MP10
Symmetry constraints		
axis	Symmetric devices	
0	MN9 - MN10; MN6 - MN7	
1	MP6 - MP7; MP1 - MP4; MP2 - MP3	
2	MP9 - MP11; MP8 - MP10; MN1 - MN2; MN3 - MN4	

Table 6: "fastcomp": Topological constraints.

uration in which high-level performance specifications can be met.

Future work includes the expansion of the numerical sensitivity computation tool to account for new parameters and analyses and its integration into the new top-down, constraint-driven circuit synthesis framework currently under development.

7 Acknowledgements

This research has been partially supported by DARPA and SRC. The first author has been supported by a grant of Asea Brown Boveri, inc. Baden, Switzerland. The authors would like to thank Umakanta Choudhury who first formalized the approach and whose suggestions were extremely useful. The authors would also like to thank Prof. R. Meyer of UC-Berkeley and Dr. A. Gandini of CSELT, Torino, Italy for providing the examples presented in this work.

References

[1] 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.

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

[3] E. Malavasi, D. Pandini and V. Liberali, "Optimum Stacked Layout for Analog CMOS ICs", in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 1711–1714, May 1993.

[4] 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.

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

[6] E. Charbon, E. Malavasi, U. Choudhury, A. Casotto and A. Sangiovanni-Vincentelli, "A Constraint-Driven Placement Methodology for Analog Integrated Circuits", in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 2821–2824, May 1992.

[7] E. Felt, E. Malavasi, E. Charbon, R. Totaro and A. Sangiovanni-Vincentelli, "Performance-Driven Compaction for Analog Integrated Circuits", in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 1731–1735, May 1993.

[8] 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.

[9] S. W. Director and R. A. Rohrer, "The generalized adjoint network and network sensitivities", *IEEE Trans. on Circuit Theory*, vol. 16, pp. 318–323, Aug 1969.

[10] D. A. Hocevar, P. Yang, T. N. Trick and B. D. Epler, "Transient sensitivity computation for MOSFET circuits", *IEEE Trans. on CAD*, vol. 4, pp. 609–620, Oct 1985.

[11] 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.

[12] J.-M. Shyu, "Performance Optimization of Integrated Circuits", Memorandum UCB/ERL M88/74, University of California at Berkeley, Nov 1988.

[13] D. J. Allstot and Jr. W. .C. Black, "Technological Considerations for Monolithic MOS Switched-Capacitor Filtering Systems", in *Proc. of the IEEE*, pp. 967–986, 1983.

[14] 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, Oct 1989.

[15] U. Choudhury, "Sensitivity Computation in SPICE3", M.s. thesis, University of California at Berkeley, 1988.