Generalized Constraint Generation in the Presence of Non-Deterministic Parasitics

Edoardo Charbon, Paolo Miliozzi *, Enrico Malavasi,
and Alberto L. Sangiovanni-Vincentelli *

Cadence Design Systems Inc., San Jose, CA
* Department of EECS, University of California, Berkeley, CA

Abstract

In a constraint-driven layout synthesis environment, parasitic constraints are generated and implemented in each phase of the design process to meet a given set of performance specifications. The success of the synthesis phase depends in great part on the effectiveness and the generality of the constraint generation process. None of the existing approaches to the constraint generation problem however are suitable for a number of parasitic effects in active and passive devices due to non-deterministic process variations. To address this problem a novel methodology is proposed based on the separation of all variables associated with non-deterministic parasitics, thus allowing the translation of the problem into an equivalent one in which conventional constrained optimization techniques can be used. The requirements of the method are a well-defined set of statistical properties for all parasitics and a reasonable degree of linearity of the performance measures relevant to design.

1 Introduction

The design of the analog section of complex mixed-signal systems is often the bottleneck of the entire design. This is due to the extreme criticality of analog designs and the difficulty to meet a set of specifications imposed upon them. The design task may be problematic if compared with digital circuits of similar complexity because of the higher number of specifications and the importance of second order effects.

To cope with this problem a number of approaches have been proposed based on the use of various performance models accounting for performance degradation due to the details of the physical implementation. Compact performance models can be achieved by linearizing a circuit around its operating point. The main requirement of such a model to be accurate is that one operates *close enough* to the operating point.

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

The constraint generation problem was formally stated in [1] and [2] for a class of parasitic effects generally associated with interconnect and generalized to all classes in [8]. In these formulations, the criticality of parasitics is quantified based on the cumulative effect to performance and used to drastically reduce the number of specifications on each interconnect realization. These approaches however postulate the deterministic nature of parasitics as a necessary condition to be meaningful. This restriction may in fact represent a serious limitation to the accuracy of performance models when technological gradients and other stochastic effects begin to dominate, thus undermining the effectiveness of constraint-based methods.
Non-deterministic parasitics mainly arise from the following sources:

- mask misalignment
- technology gradients

Although fundamentally different in nature, these errors have an impact on parasitics which can be modeled in a similar manner. A convenient way of representing parasitics of this type is through an appropriate statistical model. Assuming that a joint probability distribution function is known for all non-deterministic parasitics in the circuit, a moment generating function can be derived and a multidimensional tensor can be built of all combinations of lth order moments for each pair of parasitics. For practical purposes generally designers assume non-deterministic parasitics to be jointly Gaussian, hence uniquely characterized in terms of the mean vector and the variance-covariance matrix.

Using this representation, a compact stochastic model can be built for every performance measure. The model can then be used to map high-level specifications on the variance of the allowed performance degradation onto constraints on the tolerance for each non-deterministic parasitic component. Since a relation generally exists between parasitic tolerance and layout geometry, as shown for example in [9], it is therefore possible to generate physical constraints on the relative distances and/or orientations of the objects in the layout.

The stochastic model is then used in a constrained optimization cycle to generate a set of constraints on critical parasitics to be used in the design of circuit components.

The method's sole assumption is that the statistical properties of all parasitics be known *a priori* and that a bounded cross-correlation matrix exist. Constraints on the statistical parameters of parasitics are computed using sensitivity analysis and constrained optimization. The objective of the optimization is a continuous and bounded function, called *flexibility*, which represents an estimate of how realistic the constraint will be in an actual design. A technology-dependent model of the dependence of parasitics on the physical realization of the design is sufficient to guide a set of constraint-driven tools to meet the desired performance specifications.

The paper is structured as follows. The foundations of the constraint generation problem and its application to constraint-based synthesis systems are reviewed in section 2. The techniques used for the derivation of performance models accounting for non-deterministic parasitics are presented in section 3. The engine for the calculation

ICCAD '96
1063-6757/96 $5.00

of parasitic and topological constraints is outlined in section 4. The suitability of the method is illustrated through examples in section 5.

2 Constraint Generation: Problem Formulation

For an arbitrary circuit, let us define performance vector $\mathbf{K}$ as the finite array of all N_k measures that evaluate a parametric behavior for the circuit. Let $\mathbf{K_0}$ be the performance nominal value and $\triangle\mathbf{K}$ the degradation of performance with respect to it. Assume that all parasitics significantly affecting performance are known. Specifications are expressed as the maximum allowed performance degradation from nominal.

$$\triangle\ \mathbf{K} = \mathbf{K} - \mathbf{K_0} \leq \overline{\triangle\mathbf{K}}, \tag{1}$$

where term $\overline{\triangle\mathbf{K}}$ represents the constraint on degradation $\triangle\mathbf{K}$. Performance degradations are due to process variations and to the parasitics caused by the realization of the layout details. Both absolute parasitic values and mismatch play a role in the deviation of performance measures from nominal.

Assume that the design architecture is known *a priori* and that all relevant parasitics are defined in terms of an array $\mathbf{p}$ of size N_p. The process of mapping the constraint of equation (1) onto constraints associated with parasitic components of the type

$$\mathbf{p} \leq \mathbf{p}^{(bound)}, \tag{2}$$

is called *parasitic constraint generation.* Formally the problem is defined as follows.

Problem 1 *Given a circuit C with performance $\mathbf{K}$ and a finite set of parasitic components $\mathbf{p}$, find bounds on all parasitics, such that (1) holds.*

The solution of this problem is nontrivial for two reasons. First, performance $\mathbf{K}$ is generally an array of non-linear functions of parasitics, often not representable in a compact form. Second, the number of parasitics is generally much larger than the size of the performance array. Hence a naive approach based on solving equation (1) with respect to each parasitic is not feasible.

In order to maximize the success rate of the implementation of parasitic constraints at every phase of a constraint-based synthesis flow, the parasitic generation process can be represented in terms of the following mathematical optimization problem

$$\begin{array}{ll} & maximize:\ \ f(\mathbf{p}) \\ \text{subject to:} & (i)\quad \triangle\mathbf{K}(\mathbf{p}) \leq \overline{\triangle\mathbf{K}} \\ & (ii)\quad \mathbf{p}^{(min)} \leq \mathbf{p} \leq \mathbf{p}^{(max)} \end{array} \tag{3}$$

where function $f()$, also known as *flexibility function*, denotes the degree of difficulty required by the implementation of the circuit using $\mathbf{p}^{(bound)}$ in (2) as constraints to all parasitic components. A similar representation can be used at higher levels of design hierarchy. In this case, parasitic bounds are substituted with the performance measures associated with a given level of hierarchy and the flexibility function relates to the difficulty of meeting a bound on a class of performance measures. Such scheme has been proposed as an effective design methodology in a number of mixed-mode applications [10].

Inequality (3)i enforces all performance specifications simultaneously, while (3)ii insures that the constraints be feasible. Term $\triangle\mathbf{K}(\mathbf{p})$ in (3) represents the model of the performance degradation in terms of all circuit parasitics. A key to the efficiency of the optimization problem relates to the simplicity of this model. As an example, consider a model based on the first order Taylor expansion of $\mathbf{K}$ in terms of all parasitic components

$$\triangle\ \mathbf{K}(\mathbf{p}) = \mathbf{S}(\mathbf{p} - \mathbf{p_0}), \tag{4}$$

where $\mathbf{S}$ is the matrix of the performance sensitivities relative to all known parasitics, defined as following

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

Each entry $S_{i,j}$ in the matrix represents the sensitivity of the ith measure of $\mathbf{K}$ with respect to the jth parasitic component in $\mathbf{p}$. Vector $\mathbf{p_0}$ is the parasitic nominal value.

If performance degradations are approximated by linearized expressions using sensitivities, the approximations are acceptable provided that all degradations be small compared to the nominal values. The array of all degradations is then

$$\triangle\mathbf{K}(\mathbf{p}) \approx \mathbf{S}\left[\mathbf{p} - \mathbf{p}_0\right]. \tag{5}$$

Before the definition of layout details, one cannot take advantage of the possible cancellation effects due to positive 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. Assuming that the performance model of (5) is used, equation (1) becomes

$$\triangle\ \mathbf{K}(\mathbf{p}) - \overline{\triangle\mathbf{K}^+} \leq \mathbf{0} \tag{6}$$

$$\triangle\ \mathbf{K}(\mathbf{p}) + \overline{\triangle\mathbf{K}^-} \geq \mathbf{0}, \tag{7}$$

where $\overline{\triangle\mathbf{K}^+}$ and $\overline{\triangle\mathbf{K}^-}$ 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 (5) in inequalities (6) and (7), the general problem can be rewritten as

$$\mathbf{S}^+\left[\mathbf{p} - \mathbf{p}_0\right] - \overline{\triangle\mathbf{K}^+} \leq \mathbf{0} \tag{8}$$

$$\mathbf{S}^-\left[\mathbf{p} - \mathbf{p}_0\right] - \overline{\triangle\mathbf{K}^-} \leq \mathbf{0}, \tag{9}$$

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

$$\mathbf{S}^+_{i,j} = \max(0, S_{i,j})$$
$$\mathbf{S}^-_{i,j} = \max(0, -S_{i,j}),$$

In the remainder of this paper the '$+$' and '$-$' signs have been omitted in the notations of sensitivities and constraints. Expressions (8) and (9) are given for positive and negative directions, and the general problem formulation becomes

$$\mathbf{S}\left[\mathbf{p} - \mathbf{p}_0\right] - \overline{\triangle\mathbf{K}} \leq \mathbf{0}. \tag{10}$$

The flexibility is generally a normalized function of the j-th component p_j of $\mathbf{p}$ of the type

$$\text{flexibility}(p_j) = 1 - \frac{p_j^{(max)} - p_j}{p_j^{(max)} - p_j^{(min)}}, \tag{11}$$

where $p_j^{(max/min)}$ represent the feasibility region for parasitic p_j. This function penalizes lower values and does not reward elevated values of the constraint. Similar dedicated functions are used for other classes of parasitics.

3 Non-Deterministic Constraint Generation

In general, IC interconnect lines and integrated devices can be characterized with reasonable accuracy by relatively compact models up to a few Gigahertz. In some cases however, the modeling parameters can be determined with high uncertainty even when the process is known *a priori*. To cope with this problem, the parameters are often expressed in terms of a known statistical behavior. The process of finding constraints for such parameters must then necessarily involve the derivation of bounds on their statistical behavior.

Let array $\boldsymbol{\pi}$ of size N_π define a set of known non-deterministic parasitic components and let $F(\pi_j)$ be the array of all statistical distributions associated with $\pi_j \in \boldsymbol{\pi}$. Assume now that the first n moments exist and are bounded for all components of $\boldsymbol{\pi}$. Furthermore, assume that all components of $\boldsymbol{\pi}$ be statistically independent. Then, the first n moments $m_l(\triangle\mathbf{K})$ of degradation $\triangle\mathbf{K}$ can be computed as

$$m_l(\triangle\mathbf{K}) = \mathbf{S}_\pi^{(l)} m_l(\boldsymbol{\pi}), \tag{12}$$

where $m_l(.)$ denotes the lth moment and $\mathbf{S}_\pi^{(l)}$ a matrix whose (i,j)-entries equal the lth power of the corresponding (i,j)-entry of $\mathbf{S}_\pi$, the sensitivity matrix of $\mathbf{K}$ with respect to non-deterministic parasitic vector $\boldsymbol{\pi}$.

In order to use (3) to derive bounds on the first n moments of $\boldsymbol{\pi}$ it is sufficient to replace (3)i with the following set of constraints

$$m_l(\triangle\mathbf{K}) \leq \overline{m_l(\triangle\mathbf{K})},\ \forall l = 1, \ldots, n\ , \tag{13}$$

where $\overline{m_l(\triangle\mathbf{K})}$ denotes a constraint on the maximum acceptable value of the lth moment of $\triangle\mathbf{K}$. Moreover, (3)ii needs be substituted with the following constraint

$$m_l(\boldsymbol{\pi})^{(min)} \leq m_l(\boldsymbol{\pi}) \leq m_l(\boldsymbol{\pi})^{(max)},\ \forall l = 1, \ldots, n\ , \tag{14}$$

where $m_l(\boldsymbol{\pi})^{(min/max)}$ are estimates of the lower/upper-bound of the lth moment of $\boldsymbol{\pi}$.

In most designs however some or all the components of $\boldsymbol{\pi}$ are not statistically independent, then equation (12) needs be modified, while the reminder of the optimization problem remains unchanged.

Let $\mathbf{Q}_\pi$ be the variance-covariance matrix of $\boldsymbol{\pi}$ defined as

$$\mathbf{Q}_\pi = E(\boldsymbol{\pi}\boldsymbol{\pi}^T), \tag{15}$$

where operator $E(.)$ indicates the expected value. By definition $\mathbf{Q}_\pi$ is a symmetric positive-definite $N_\pi \text{x} N_\pi$ square matrix. Consequently, it can be decomposed as

$$\mathbf{Q}_\pi = \mathbf{U^T D U}, \tag{16}$$

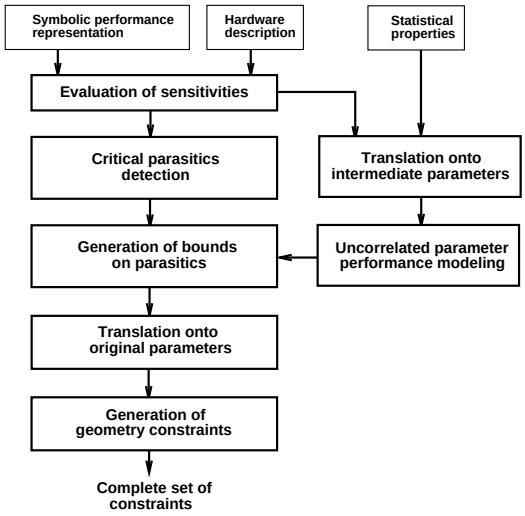

Figure 1: *Flow diagram of the constraint generation process*

where $\mathbf{U}$ is a $N_\pi \text{x} N_\pi$ non-singular mapping and $\mathbf{D}$ a diagonal positive definite matrix. One can easily show that if $\boldsymbol{\pi}$ is mapped onto vector $\mathbf{y} = \mathbf{U}\boldsymbol{\pi}$, the new parameters in $\mathbf{y}$ are uncorrelated, i.e. the variance-covariance matrix $E(\mathbf{y}\mathbf{y}^T)$ is diagonal. Alternatively the original vector $\boldsymbol{\pi}$ can be obtained as

$$\boldsymbol{\pi} = \mathbf{U^{-1}y} = \mathbf{U^T y}, \tag{17}$$

where relation $\mathbf{U}^{-1} = \mathbf{U}^T$ is the consequence of the fact that Q_π is positive definite and symmetric. If the parameters in $\boldsymbol{\pi}$ are Gaussian random variables, then the entries of $\mathbf{y}$ are statistically independent. Hence, the model of equation (12) can be used substituting $\boldsymbol{\pi}$ with $\mathbf{U^T y}$.

To insure meaningful results, bounds $y_j^{(bound)}$, obtained as the solution of problem (3), need be translated onto the corresponding ones associated with $\boldsymbol{\pi}$ using the transformation of equation (17). For $l = 2$ the procedure yields

$$m_2(\boldsymbol{\pi})^{(bound)} = \mathbf{U}^T m_2(\mathbf{y})^{(bound)} \mathbf{U}. \tag{18}$$

4 Constraint Computation Engine

The flow diagram of the constraint generation process is shown in Figure 1. From hardware and performance description, sensitivity analysis of the circuit is performed by means of standard mixed symbolic or numerical techniques [11, 12]. A performance model based on sensitivities is built accounting for all parasitics

$$\triangle\,\mathbf{K}(\mathbf{p}, \boldsymbol{\pi}) = \mathbf{S}_\pi(\boldsymbol{\pi} - \boldsymbol{\pi_0}) + \mathbf{S}(\mathbf{p} - \mathbf{p_0}), \tag{19}$$

where $\boldsymbol{\pi}_0$ is the mean of $\boldsymbol{\pi}$. Using the variance-covariance information associated with all non-deterministic parasitics, $\mathbf{U}$ is derived by means of Singular Value Decomposition. The intermediated parameter vector $\mathbf{y}$ is computed and the performance model is modified as following

$$\triangle\,\mathbf{K}(\mathbf{p}, \mathbf{y}) = \mathbf{S_y}(\mathbf{y} - \mathbf{y_0}) + \mathbf{S}(\mathbf{p} - \mathbf{p_0}), \tag{20}$$

where the modified sensitivity matrix $\mathbf{S_y}$ and mean value $\mathbf{y_0}$ are derived as

$$\mathbf{S_y} = \mathbf{S}_\pi \mathbf{U^T},\ \mathbf{y_0} = \mathbf{U}\boldsymbol{\pi_0}. \tag{21}$$

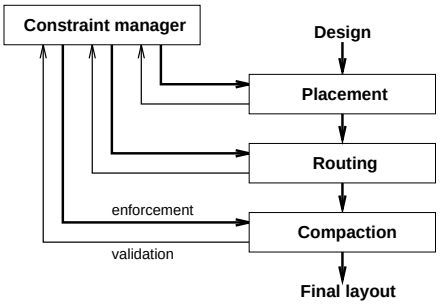

Figure 2: *Constraint management within a constraint-based system*

For simplicity but without loss of generality assume that $\mathbf{y_0} = \mathbf{0}$ and $\boldsymbol{\pi_0} = \mathbf{0}$. Using the extended performance model and *a priori* estimates of upper-bounds on parasitics $\mathbf{y}^{(\mathbf{max})}$, the subset of N_c' critical parasitics is detected by eliminating all parasitics for which

$$\sum_{j=1}^{N_\pi - N_c'} |S_{y_\pi,j}|^l m_l(y_j^{(max)}) < \alpha \overline{m_l(\triangle K_i)}, \quad \forall l = 1, \ldots, n;\ i = 1, \ldots, N_k \tag{22}$$

is satisfied. In our approach term α was chosen to be 1%. Deterministic parasitics are treated similarly, where the condition for finding N_c" critical parasitics is the following

$$\sum_{j=1}^{N_p - N_c"} S_{i,j} p_j^{(max)} < \alpha \overline{\triangle K_i},\ \forall l = 1, \ldots, n;\ i = 1, \ldots, N_k\ . \tag{23}$$

A quadratic programming approach [3] is used to compute the final bounds on critical parasitics and on the first n moments of the intermediate parameters. Finally, all intermediate parameter bounds are reconverted into bounds on the original parameters using equation (17). As a post-processing step, all constraints may undergo a technology mapping phase targeted towards computing detailed geometrical bounds on all layout components, thus satisfying all specifications on the yield properties of the final circuit. The constraint generation process is the core of our constraint-driven layout synthesis paradigm presently under development. The translation routines are embedded in a constraint manager which is intended to serve all the existing and future constraint-driven tools within the design environment.

The design need first be partitioned by the designer into functional units for which a set of performance measures is defined. Using budgeting techniques such as those proposed in [10], constraints are computed for each performance measure. Each module is subpartitioned into simpler units until the design has been fully traversed and performance constraints are mapped onto a set of parasitic constraints. At this point of the design the layout assembly and hence the constraint enforcement process takes place.

Figure 2 shows the constraint manager and its interaction with the constraint-based tools. In every phase of the layout the constraints are generated *on demand*, i.e. the set of all constraints is partitioned into subsets with all the constraints needed by a particular layout phase. The constraint generator operates on the subsets *incrementally*, i.e. calculating all subset constraints while using extracted values for parasitics of previous phases and estimates for parasitics of future phases. Hence problem (3) is modified as

$$\begin{aligned} &maximize:\ f(\mathbf{p}) \\ \text{subject to:}\quad &(i)\quad \triangle\mathbf{K}(\mathbf{p}) \le \overline{\triangle\mathbf{K}} - \triangle\mathbf{K}^{(future)} \\ &(ii)\quad \mathbf{p}^{(min)} \le p \le \mathbf{p}^{(max)} \\ &(iii)\quad p_j = p_j^{(ext)},\ \forall j \in \mathcal{P}_{prev} \end{aligned} \tag{24}$$

where the terms $p_j^{(ext)}, \forall j \in \mathcal{P}_{prev}$ represent extracted parasitics associated with layout structures which have already been generated. Term $\triangle\mathbf{K}^{(future)}$ on the contrary relates to the budgeted performance degradation due to layout structures yet to be generated.

The enforcement is followed by a constraint validation phase to verify whether or not the enforcement process was successful, thus presenting to the user a number of alternative strategies in case of failed validation. This is done fully interactively to allow the user to gradually guide the design to a satisfactory result. Both the budgeting and constraint translation phases are applied incrementally while the design unfolds. Editing and local rebudgeting are also allowed.

5 Results

The algorithms outlined in this paper have been tested on a wide range of fully analog and switched circuits, selected from a set of commonly used industrial applications. We will discuss here two circuits to highlight the features of the proposed algorithms.

For the computation of all sensitivity measures, standard analysis methods available in SPICE have been used. Presently available are dc, ac and transient analysis. The set of design parameters currently supported are *resistances, substrate capacitances, capacitive coupling, inductances, mutual inductances, threshold voltage, channel length and width* for CMOS ICs. Additionally, *base-emitter junction area* sensitivities can be computed for BiCMOS ICs.

Class AB amplifier

Consider power amplifier "ab", depicted in Figure 3. The performance measures analyzed are unity gain bandwidth and low frequency

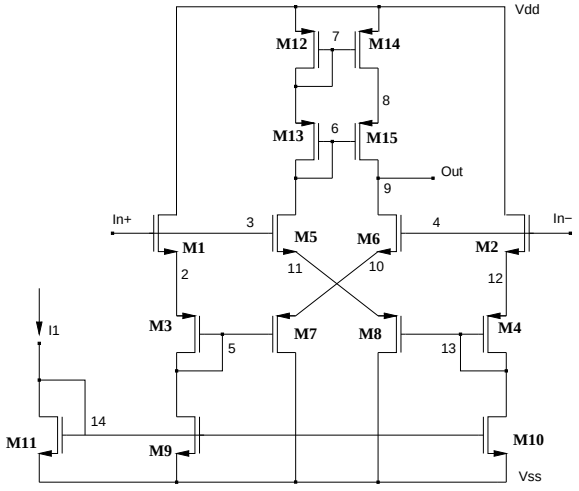

Figure 3: *Class AB amplifier "ab"*

Performance	Nominal	$\overline{\triangle \mathbf{K}^-}$	$\overline{\triangle \mathbf{K}^+}$	$\overline{m_2(\triangle \mathbf{K})}$
Unity gain bandwidth	25.0 MHz	-5 MHz	∞	$(1\%)^2$
Low frequency gain	51.0 dB	-3 dB	∞	$(1\%)^2$
Offset Voltage	14 mV	-2 mV	2 mV	∞
Phase margin	106 deg	-3deg	3deg	∞

Table 1: "ab": Nominal performance and specifications

Type	# Critical parasitics
Resistance	12
Resistive mismatch	13
Capacitance/cross-coupling	188
Capacitive mismatch	27

Table 2: "ab": Constraints on critical deterministic parasitics

Deterministic constraints	Non-deterministic constraints
275 sec	1,116 sec

Table 3: "ab": CPU times for the computation of all constraints

Performance	Nominal	$\overline{\triangle \mathbf{K}^-}$	$\overline{\triangle \mathbf{K}^+}$	$\overline{m_2(\triangle \mathbf{K})}$
Unity Gain Bandwidth	3.0 MHz	-0.3	∞	$(1\%)^2$
Low Frequency Gain	120 dB	-3	∞	$(1\%)^2$
Phase Margin	60°	-1°	∞	∞

Table 4: "mph": Nominal performance and specifications

Type	# Critical parasitics
Resistance	40
Resistive mismatch	4
Capacitance/cross-coupling	153
Capacitive mismatch	51

Table 5: "mph": Constraints on critical deterministic parasitics

Deterministic constraints	Non-deterministic constraints
12,129 sec	5,112 sec

Table 6: "mph": CPU times for the computation of all constraints

gain, phase margin, and offset voltage. Table 1 reports nominal values, positive and negative constraints on the degradation of each performance. Table 2 summarizes the results of deterministic parasitic constraints computed for capacitive and resistive parasitics using the techniques described in section 2. The CPU times for the computation of all constraints are listed in Table 3 for a DEC AlphaServer 2100 5/250 . Table 7 lists the constraints on the most critical non-deterministic parasitics. The size of the original variance-covariance matrix was 45x45. The tabulate can be directly used for the computation of the maximum distance allowed between objects whose constraints on the maximum cross-correlation is less than infinity.

Low power amplifier

To conclude, consider the low power amplifier "mph", depicted in Figure 4. This circuit is of particular interest because of the presence of several feedback and forward paths insuring the stability of the circuit. Another point of interest is the low supply required by the circuit, making it particularly sensitive to all device threshold voltages. Table 4 lists the performance measures of interest for this circuit and related constraints. A summary of all calculated constraints on deterministic parasitics is shown in Table 5. Table 8 shows a section of the 153x153 bound on the variance-covariance matrix. The CPU times required for all constraint derivations is reported in Table 6 for a DEC AlphaServer 2100 5/250 .

6 Conclusions

A novel constraint generation methodology has been proposed for non-deterministic parasitics to be applied in constraint-driven layout synthesis flows. The method is based on the separation of variables associated with non-deterministic parasitics, thus allowing the translation of the problem into an equivalent one which can be solved using conventional constrained optimization techniques. The requirements of the method are a well-defined set of statistical properties for all parasitics and a reasonable degree of linearity of the performance measures relevant to the design.

References

[1] U. Choudhury and A. L. 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] 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*, volume 4, pp. 2008–2011, June 1991.

[3] U. Choudhury and A. L. Sangiovanni-Vincentelli, "Constraint Generation for Routing Analog Circuits", in *Proc. IEEE/ACM DAC*, pp. 561–566, June 1990.

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

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

[6] E. Malavasi, E. Felt, E. Charbon and A. L. Sangiovanni-Vincentelli, "Symbolic Compaction with Analog Constraints", *International Journal of Circuit Theory and Applications, Special Issue on "Analog Tools for Circuit Design", John Wiley & Sons*, vol. 23, n. 4, pp. 433–452, July-August 1995.

[7] E. Malavasi, E. Charbon, E. Felt and A. L. Sangiovanni-Vincentelli, "Automation of IC Layout with Analog Constraints", *IEEE Trans. on CAD*, vol. 15, n. 8, pp. 923–942, August 1996.

[8] E. Charbon, E. Malavasi and A. L. Sangiovanni-Vincentelli, "Generalized Constraint Generation for Analog Circuit Design", in *Proc. IEEE ICCAD*, pp. 408–414, November 1993.

[9] 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. SC-24, pp. 1433–1440, October 1989.

[10] H. Chang, A. L. 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 CICC*, pp. 841–846, May 1992.

[11] D. A. Hocevar, P. Yang, T. N. Trick and B. D. Epler, "Transient Sensitivity Computation for MOSFET Circuits", *IEEE Trans. on CAD*, vol. CAD-4, n. 4, pp. 609–620, October 1985.

[12] S. W. Director and R. A. Rohrer, "The Generalized Adjoint Network and Network Sensitivities", *IEEE Trans. on Circuit Theory*, vol. CT-16, pp. 318–323, August 1969.

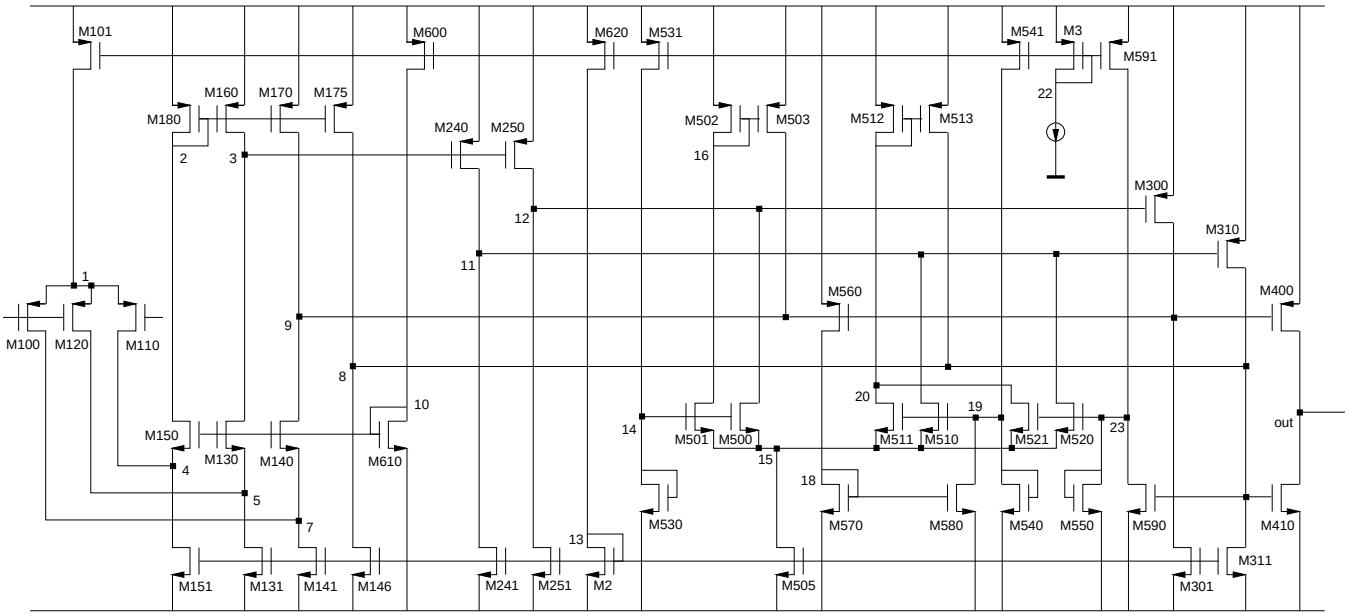

Figure 4: *Low power amplifier "mph"* (Courtesy of R.G.H. Eschauzier and J. H. Huijsing, TU Delft, The Netherlands)

	L_{m1}	W_{m1}	VTO_{m1}	L_{m2}	W_{m2}	L_{m5}	W_{m5}	L_{m6}	W_{m6}
L_{m1}	$1.968E-4$	$5.254E-7$	$2.355E-7$	∞	∞	∞	∞	∞	∞
W_{m1}	$5.254E-7$	$1.924E-4$	$7.786E-6$	$3.699E-6$	$4.504E-06$	$2.091E-6$	$1.337E-6$	$7.299E-7$	$7.601E-8$
VTO_{m1}	$2.355E-7$	$7.785E-6$	$1.232E-4$	∞	∞	∞	∞	∞	∞
L_{m2}	∞	$3.699E-6$	∞	$1.617E-4$	$1.635E-5$	∞	∞	$2.213E-6$	∞
W_{m2}	∞	$4.504E-6$	∞	$1.635E-5$	$1.494E-4$	∞	$1.478E-5$	$2.897E-6$	∞
L_{m5}	∞	$2.091E-6$	∞	∞	∞	$1.601E-4$	$9.412E-6$	∞	$7.118E-6$
W_{m5}	∞	$1.337E-6$	∞	∞	$1.478E-5$	$9.412E-6$	$1.636E-4$	$2.334E-6$	∞
L_{m6}	∞	$7.299E-7$	∞	$2.213E-6$	$2.897E-6$	∞	$2.334E-6$	$1.514E-4$	$9.459E-6$
W_{m6}	∞	$7.601E-8$	∞	∞	∞	$7.118E-6$	∞	$9.459E-6$	$1.459E-4$

Table 7: "ab": Bounds on the acceptable variance-covariance matrix

	L_{M3}	W_{M3}	VTO_{M3}	L_{M100}	W_{M100}	VTO_{M100}	L_{M110}	W_{M110}	VTO_{M110}
L_{M3}	$9.761E-5$	$9.606E-5$	$3.059E-6$	$1.457E-6$	$1.789E-6$	$3.282E-6$	$2.467E-6$	$2.793E-6$	$2.767E-6$
W_{M3}	$9.606E-5$	$1.012E-4$	$1.499E-6$	$2.327E-6$	$2.333E-6$	$2.342E-6$	$2.336E-6$	$2.342E-6$	$2.342E-6$
VTO_{M3}	$3.059E-6$	$1.499E-6$	$7.997E-5$	∞	∞	$8.471E-6$	$5.338E-6$	$7.955E-6$	$7.087E-7$
L_{M100}	$1.457E-6$	$2.327E-6$	∞	$1.457E-4$	$1.273E-6$	$2.651E-6$	∞	$7.158E-6$	∞
W_{M100}	$1.789E-6$	$2.333E-6$	∞	$1.273E-6$	$1.107E-4$	$2.563E-6$	∞	$2.996E-6$	$5.599E-6$
VTO_{M100}	$3.282E-6$	$2.342E-6$	$8.471E-6$	$2.651E-6$	$2.563E-6$	$1.673E-4$	$1.521E-6$	∞	$6.245E-7$
L_{M110}	$2.467E-6$	$2.336E-6$	$5.338E-6$	∞	∞	$1.521E-6$	$1.623E-4$	$1.014E-5$	$5.788E-8$
W_{M110}	$2.793E-6$	$2.342E-6$	$7.955E-6$	$7.158E-6$	$2.996E-6$	∞	$1.014E-5$	$1.269E-4$	∞
VTO_{M110}	$2.767E-6$	$2.342E-6$	$7.087E-7$	∞	$5.599E-6$	$6.245E-7$	$5.788E-8$	∞	$1.327E-4$

Table 8: "mph": Bounds on the acceptable variance-covariance matrix