Efficient Analog Circuit Synthesis
with simultaneous Yield and Robustness Optimization

Geert Debyser, Georges Gielen*
Katholieke Universiteit Leuven, Belgium
[geert.debyser, georges.gielen]@esat.kuleuven.ac.be

Abstract
This paper presents an efficient statistical design methodology that allows simultaneous sizing for performance and optimization for yield and robustness of analog circuits. The starting point of this methodology is a declarative analytical description of the circuit. An equation manipulation program based on constraint satisfaction converts this declarative model into an efficient design plan for optimization based sizing. The efficiency is due to the use of an operating point driven DC formulation, so that the design plan avoids the calculation of simultaneous sets of nonlinear equations. From the same declarative analytical description also a direct symbolic yield estimation plan is generated. The parametric yield is estimated by propagating the spread of the technological variables through the analytical model towards the performance variables of the circuit. The design plan and the yield estimation plan are then combined together in the inner loop of a global optimization routine. The strength of this methodology lies in the low CPU times needed to perform yield estimation compared to the hours of simulation batches with Monte Carlo simulations, while the accuracy is comparable.

I. Introduction
Designing an analog circuit is one thing, producing it is another. Real-life technology parameter variations make the circuits fail for some or all of the specifications if no precautions are taken. The ratio of the number of successful circuits over the number of produced circuits is the total yield. The total yield consists of yield due to production faults and yield due to soft faults. In this paper we concentrate on the yield based on the soft faults, generated by the technology parameter variations: the parametric yield.

The hard way to make the design more robust is to run multiple batch jobs of Monte Carlo simulations in the inner loop of an optimization routine. Because a Monte Carlo simulation consists of typically hundreds of SPICE simulations, the computational effort is so large that only a post-design yield optimization is considered. This is the known design centering method. Starting from the nominal design a local optimization tries to push the performances away from the specification boundaries in order to make the circuit more robust against technology parameter variations [1]. The same is true for analog circuit sizing or synthesis programs [2]. Most of them concentrate on the nominal design only, without considering the process parameter or operating condition variations. Performing hours of simulated annealing for just a nominal design is only half of the game. Only in [3] a first approach towards analog synthesis for manufacturability was presented that combines nominal circuit optimization with variation analysis in an outer optimization loop. The results of the variation analysis are used to change the cost function of the inner circuit optimization by penalizing design solutions that do not meet the specifications over the entire operating range. The approach however is extremely time consuming.

Therefore, in this paper, an alternative approach is presented towards simultaneous circuit sizing for performance and yield/robustness optimization. The approach results in a drastic reduction of the required CPU time, without sacrificing too much accuracy. It is based on the use of symbolic techniques to capture the behavior of a circuit in a declarative model [4]. Constraint satisfaction techniques implemented in the tool DONALD [5] are then used to derive an efficient sizing plan as well as an efficient yield estimation plan. Both plans are then simultaneously evaluated in the inner loop of a global optimization routine. The outcome of the optimization is a circuit design point that fulfills all specifications and that at the same time has pushed away the performances from the specification boundaries under the influence of the yield and/or Cpk measure.

The paper is organized as follows. Section 2 explains the construction of the sizing plan by the DONALD tool starting from the declarative behavioral model of the circuit. Section 3 describes how we used symbolic techniques to construct the yield/robustness estimation plan. Section 4 explains the yield optimization strategy. Section 5 shows experimental results. Section 6 draws some conclusions.

II. Sizing plan
A declarative analytical model of an analog circuit is mainly obtained by 2 sources: by applying symbolic methods on the circuit’s graph topology and by hand. A simple illustrative example of the outcome for an inverter is given in Fig. 1, where 7 equations describe the DC operating point.

The 7 equations are presented in a bipartite graph form (see

Fig. 1. Inverter and its simplified behavioral DC model.)
Fig. 2. Depending on the input variables, DONALD has to solve a 5x5 cluster or 2 simple equations.

Fig. 2a), where the square boxes represent the equations and the circles represent the variables. We assume that the user can give a value for $V_{dd}$, $V_{ss}$, $V_{in}$ (typically half-way $V_{dd}$ and $V_{ss}$). $L_n$ and $L_p$ can be chosen to be minimal size. The equation manipulation tool DONALD propagates these inputs through the graph, which is then partially directed. The traditional SPICE-like inputs (choosing $W_p$ and $W_n$, see Fig. 2b), lets DONALD simultaneously solve a set of 5 (possibly nonlinear) equations with a nonlinear root solver. In this case convergence cannot be guaranteed and the solution also depends on the starting point. The operating point driven DC formulation [6], however, constrains node voltages and branch currents (choosing $V_{nout}$ and $I_{dip}$, as in Fig. 2c). The DONALD program only has to solve 2 one-dimensional nonlinear equations to obtain $W_n$ and $W_p$, which can be executed much faster. The computational effort is drastically reduced since, for a circuit with $N$ nodes and $M$ MOS-transistors, solving $M$ one-dimensional nonlinear equations is of the order $O(M \text{ Root}(1))$ (where Root($x$) is the effort to solve a cluster of size $x$). A simulator needs an effort of the order $O(\text{Root}(N))$, with a risk of divergence. This approach has already been used for optimization based nominal sizing [6], [7]. In this paper we extend this to yield optimization.

III. Yield/robustness estimation plan

Our approach replaces the Monte Carlo simulations with a direct yield estimation technique. We start with a statistical transistor model, which gives us a reduced set of quasi-independent technology parameters $\theta$. Then we calculate the nominal design point and then calculate the variances of all performance parameters $y$ with respect to the reduced set of technology parameters $\theta$. Using these variances, we construct an efficient representation of "yield" based on $C_p/C_{pk}$ indices. Both the sizing plan and the yield estimation plan are then placed in the inner loop of the optimization routine as explained in section 4. The flow diagram in Fig. 3 with the setup for the yield estimation plan is explained in the following subsections ($x$ stands for designable parameters and $e$ stands for simulator variables).

A. Statistical transistor model

The default transistor models have to be replaced by statistical models in order to take the correlations of the technology parameters into account. This is necessary to estimate the yield in a statistically correct way. The statistical model describes all technological variables as a function of only 7 quasi-uncorrelated input variables, which are TOX, NSUB, $C_{ij}$, $C_{p}$, $L_d$, $L_n$, and $R_{SH}$. A method for deriving such a statistical transistor model has been discussed in [8]. The Monte Carlo routine perturbs only these 7 parameters and extracts performances from the SPICE output to construct the yield estimation figures based on a pass/fail mechanism. This way of calculating the yield is very costly: a large number (e.g. 300) of SPICE simulations have to be executed. Our symbolic yield estimator also starts from the same 7 parameters $\theta$.

B. Direct yield estimation

As the pass/fail measure used in Monte Carlo simulation is very rough and gives little or no information about which performance does not meet its specification, another quality measure is preferred. The Taguchi quality measure gives a much better idea of the quality of a circuit, because it takes absolute variability and bias to the target spec into account:

$$M_{\text{TAG}} = \text{var}[y(x)] + (\tau - \text{Spec}^T)^2$$

To ease the calculation of $M_{\text{TAG}}$, two capability indices were introduced in [9]: the capability potential index $C_p$:

$$C_p = \frac{\text{Spec}^U - \text{Spec}^L}{6\sigma_y}$$

and the capability performance index $C_{pk}$:

$$C_{pk} = \min\left\{\frac{\text{Spec}^U - \tau}{3\sigma_y}, \frac{\text{Spec}^L - \tau}{3\sigma_y}\right\}$$

The first index represents the variability, the second index the bias. As can be noticed from formulae (2,3), these indices strongly depend on the variances of the performances. The variances of the 7 technological parameters are propagated through the computational path by means of sensitivities. In case $\theta_1, \theta_2, \ldots, \theta_{7,7}$ are not correlated, the variability of the performances can be written as follows:

$$\sigma_{y_i} = \sqrt{\text{var}(y_i)} \approx \sqrt{\sum_{j=1}^{7}(S_{ij})^2\sigma_{\theta_j}^2}$$

(assuming that $\sigma_{\theta_j}, /\theta_j$ is sufficiently small).
An extension has been made to DONALD (see Fig. 4), so that it also builds a computational plan to calculate all sensitivities from the output variables w.r.t. the input variables. The sensitivities have to be known for every operating point \( x \). Our Jacobians are in symbolic form, so the Jacobian updating for every \( x \) is relatively cheap.

This computational path is derived from the same set of equations used to determine the nominal point, but \( W \) and \( L \) must be chosen as input variables. This time however, we do not solve the set of nonlinear equations, but we reuse the values previously obtained in the nominal point calculation (see Fig. 4). Symbolic derivatives from all equations are automatically derived. Out of this a computational path results, which is a chain of one-dimensional and more-dimensional subsystems of equations. The local sensitivities are calculated using Jacobians in symbolic form. The global sensitivity matrix \( S_y^g \) with elements \( S_{ji}^g = \partial y_i / \partial \theta_j \), is calculated by applying the chain rule according to the computational plan:

\[
S_y^g = S_y^s S_{yin}^s \ldots S_{yout}^s
\]

where \( z_k \) are the internal variables along the calculation path.

The calculation of the local sensitivities is complicated but straightforward. First, at each square subsystem with equations \( f_1, \ldots, f_n \), a distinction is made between those variables of the subsystem that have been solved by the subsystem, and those variables that are solved by previous subsystems. The former are called the output variables \( \mathbf{z}^{\text{out}} = (z_1^{\text{out}}, \ldots, z_q^{\text{out}})^T \) of the subsystem, the others are called the input variables \( \mathbf{z}^{\text{in}} = (z_1^{\text{in}}, \ldots, z_p^{\text{in}})^T \). For each square subsystem the change \( \delta \mathbf{z}^{\text{out}} \) is then calculated of the output variables with respect to a unit change \( \delta \mathbf{z}^{\text{in}} \) of the input variables. The local sensitivity \( S_{ji}^{z_k} = \partial z_k / \partial \theta_j \) is solved from the following system of linear equations:

\[
\nabla \mathbf{F}^{\text{out}} \delta \mathbf{z}^{\text{out}} = - \nabla \mathbf{F}^{\text{in}} \delta \mathbf{z}^{\text{in}}
\]

where \( \delta \mathbf{z}^{\text{in}} = 1 \) and

\[
\nabla \mathbf{F}^{\text{out}} = \begin{bmatrix}
\frac{\partial f_1}{\partial z_1} & \cdots & \frac{\partial f_1}{\partial z_p} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_q}{\partial z_1} & \cdots & \frac{\partial f_q}{\partial z_p}
\end{bmatrix}
\]
After optimization for the performances were obtained. This yield model was directly used in the inner loop of a yield optimization routine to perform simultaneous nominal and yield/robustness optimization. The experimental results show that the model is accurate enough to steer the optimization in the direction of a “better” design in a much faster way than using a simulator. Research is still to be continued to improve the yield model.

VI. Conclusions

A new statistical design method using symbolic techniques has been presented. A direct yield estimation model has automatically been derived from the whole set of symbolic design equations by constructing a computational plan to calculate symbolically the sensitivities of the performances w.r.t. the technology parameters. By propagating the variances of 7 quasi-independent technology parameters the quality measures $C_p$ and $C_{pk}$ for the performances were obtained. This yield model was directly used in the inner loop of a yield optimization routine to perform simultaneous nominal and yield/robustness optimization. The experimental results show that the model is accurate enough to steer the optimization in the direction of a “better” design in a much faster way than using a simulator. Research is still to be continued to improve the yield model.

Acknowledgments

The authors acknowledge the financial support for parts of this work from the Flemish IWT under the MEDEA SADE project.

References