

Chapter 2

Automatic Synthesis of Analog Integrated Circuits Including Efficient Yield Optimization

Lucas C. Severo, Fabio N. Kepler and Alessandro G. Girardi

Abstract In this chapter, the authors show the main aspects and implications of automatic sizing, including yield. Different strategies for accelerating performance estimation and design space search are addressed. The analog sizing problem is converted into a nonlinear optimization problem, and the design space is explored using metaheuristics based on genetic algorithms. Circuit performance is estimated by electrical simulations, and the generated optimal solution includes yield prediction as a design constraint. The method was applied for the automatic design of a 12-free-variables two-stage amplifier. The resulting sized circuit presented 100 % yield within a 99 % confidence interval, while achieving all the performance specifications in a reasonable processing time. The authors implemented an efficient yield-oriented sizing tool which generates robust solutions, thus increasing the number of first-time-right analog integrated circuit designs.

2.1 Introduction

Analog integrated circuit (IC) design presents different characteristics from its digital counterparts in terms of number of devices, design methodologies, and design automation.

As digital electronic systems are modeled using hardware description languages (HDLs), digital design processes are largely removed from technology considerations and from actual physics of the devices. Digital IC design typically focuses on logical correctness, maximization of circuit density, placement, and routing of

L.C. Severo · F.N. Kepler · A.G. Girardi (✉)
Alegrete Technology Campus, Federal University of Pampa, Alegrete-RS, Brazil
e-mail: alessandro.girardi@unipampa.edu.br

L.C. Severo
e-mail: lucas.severo@unipampa.edu.br

F.N. Kepler
e-mail: fabio.kepler@unipampa.edu.br

circuits. The highly automated process produces variable, fab-independent netlists and easily generated layouts that are usually “right the first time.”

Analog/mixed-signal designs, on the contrary, are notorious for requiring more than just one prototyping cycle. They typically include a wide variety of primitive devices, such as digital MOS and mid- and high-voltage MOS and bipolar junction transistors (BJT), as well as a host of passive elements that include capacitors, resistors, inductors, varactors, and diodes. These devices are often required to operate under unfriendly environments, where they have to cope with high-temperature differences, high voltages, switching noise, and interference from neighboring elements. Analog circuits are also much more sensitive to noise than digital circuits, which can severely affect their performance.

Unlike digital designs, in which only a few device parameters, such as threshold voltage, leakage, and saturation currents, need to be considered, analog/mixed-signal design must cope with much more complex specifications. It is more concerned with the physics of devices. Parameters such as voltage gain, matching, power dissipation, and output resistance, for example, depend on voltage levels, device dimensions, and process parameters. Each device in the analog world must, therefore, be carefully characterized and modeled across a very large parameter space to allow for a reliable circuit design. This process usually leads to fab-specific designs that typically require more than one iteration to get the mask set right [14].

The success for first-time-right analog design in a traditional design flow can be summarized in three parts. The first is the experience of the design team. It can be acquired with a library of tried and tested design topologies and well-characterized devices, allowing a correct estimation of process variations. The second is the availability of good design kits and device models, providing an accurate characterization of transistor behavior in different operation points. Furthermore, a wide range of statistical models must be made available, including worst-case models, statistical corner models, and Monte Carlo mismatch models, making it possible for circuit design sizing and design centering techniques to achieve high-yielding and robust designs. The third is a good planning for optimizing the time necessary for a full design cycle, from initial specification to a functional prototype. A tight time-to-market, in general, plays a crucial role in the definition of the design schedule. In this context, a mandatory strategy for first-time-right silicon in analog design is the automation of critical design stages such as transistor sizing.

Automatic synthesis of analog integrated circuits is a very hard task due to the complex relationship between technology process parameters, device dimensions, and design specifications. The design of analog building blocks requires circuit parameters to be sized such that design specifications are met or even optimized. An efficient search in the design space is mandatory when hard specifications must be accomplished, mainly for low-voltage and low-power design. The exploration of all transistor operation regions is also fundamental for the search for an optimized circuit [18].

As devices shrink with the fabrication technology evolution, the impact of process variations on analog design becomes significant and can lead to circuit performance degradation and yield falling below specification [1, 20, 35]. Gate

oxide thickness, for example, approaches a few angstroms in state-of-the-art technologies. Although V_{DD} scales to sub 1 V supply voltage headroom, threshold voltage does not scale in the same proportion due to leakage. Less headroom means more sensitivity to threshold voltage variation. This issue has led to the inclusion of yield prediction as a fundamental step in the analog design process. However, this prediction—estimated by Monte Carlo analysis—might demand high computational effort if included at each iteration of an optimization procedure [36].

With Monte Carlo simulation, one can find out how the distribution in circuit response relates to the specification. The aspects of yield considered here are the percentage of devices, which meet the specification and the design centering with respect to the specification.

Another important aspect is avoiding over-design, when the circuit characteristics are within specification but with a wide margin, which could be at the expense of area or power and ultimately, cost. Although not recommended, this strategy is still in use by most of the analog design teams because of the low level of design automation.

A typical design process is iterative, first for finding a solution which meets the nominal specification, and then moving on to a solution that meets yield and economic constraints, including the effects of variations in device characteristics. It helps to understand the relationship of the design parameters to the circuit response and the relationships of the different types of circuit response. However, it is a slow process, since it depends on the direct influence of the human designer.

The inclusion of yield prediction in the automatic circuit sizing procedure allows for a realistic modeling which contributes for a first-time-right design. The problem is that it often presents high complexity due to the long simulation time in the optimization process. Several hours is often required to optimize a typical-sized circuit.

In this work, we demonstrate the main aspects and implications of automatic sizing including yield. Different strategies for accelerating performance estimation are addressed.

In Sect. 2.2, we show the different strategies for automatic analog circuit sizing, while in Sect. 2.3 we show how it can be approached as an optimization problem. In Sect. 2.4, we present a tool for circuit sizing using optimization and considering yield. Then, in Sect. 2.5, we discuss the results of the automatic design of a Miller OTA using the tool. Finally, we draw conclusions in Sect. 2.6.

2.2 Strategies for Automatic Analog Integrated Circuit Sizing

In the analog design flow, the definition of transistor sizes, device values, and bias voltages and currents is called the sizing procedure. It can be implemented, in general, by two approaches: knowledge-based sizing or optimization-based sizing.

In the knowledge-based approach, the circuit sizing is performed based on the experience of the design team. This method uses analytic design equations that relate circuit performance to device characteristics. Although it is a good approach for older technologies, it is not suitable for designs in modern fabrication technologies, since the modeling of short-channel effects turns the design equations extremely complex and simplification leads to values far from the actual circuit response. Also, it is difficult to explore transistor operation regions other than strong inversion. An example of knowledge-based sizing tool is Procedural Analog Design (PAD) tool [33].

The optimization-based approach transforms the design procedure in a general optimization problem. The circuit performance is modeled in a cost function, and the design space is explored automatically by an optimization heuristic in the search for optimized solutions. According to Barros et al. [5], the optimization method is dependent on the design optimization model, which can be classified in equation based, simulation based, or learning based.

The equation-based method uses simplified equations originated from large- and small-signal analysis of the circuit topology. It allows for a fast estimation of circuit performance, but lacks in accuracy. The application of this method has been demonstrated in the literature, mainly with the use of geometric programming [15, 23]. The circuit performance is modeled by posynomial equations, which guarantee the finding of an optimal solution in a fast computational time. However, this modeling implies simplifications that compromise accuracy, since performance equations are not posynomials.

Simulation-based methods use electrical simulators such as SPICE to estimate circuit performance. This performance estimation method is purely numerical and tends to consume a large computational time, since several iterations are necessary to resolve the convergence algorithm implemented by SPICE. However, this method gives a very accurate performance estimation. Electrical simulation allows the calculation of all design specifications, in both time and frequency domains. Another advantage is that circuit variability and sensitivity can be estimated by corner models or Monte Carlo simulation.

The tool proposed by Phelps et al. [29] uses simulated annealing heuristic to explore a multi-objective cost function using Cadence Spectre simulator for performance estimation. The exploration of the design space using multi-objective genetic optimization is presented by De Smedt and Gielen [8], in which the calculation of the hypersurface of Pareto-optimal design points explores the trade-off between competing objectives.

Learning-based methods provide fast performance evaluation and good accuracy. It is obtained by using techniques such as support vector machines [5] and neural fuzzy networks [3]. The models are trained from electrical simulations. The drawbacks are the high effort necessary to train the models with the desired accuracy—a huge amount of simulation data is necessary—and the low configurability, since a simple modification in the circuit topology makes the trained model no more suitable for the application.

Fig. 2.1 General optimization-based design flow

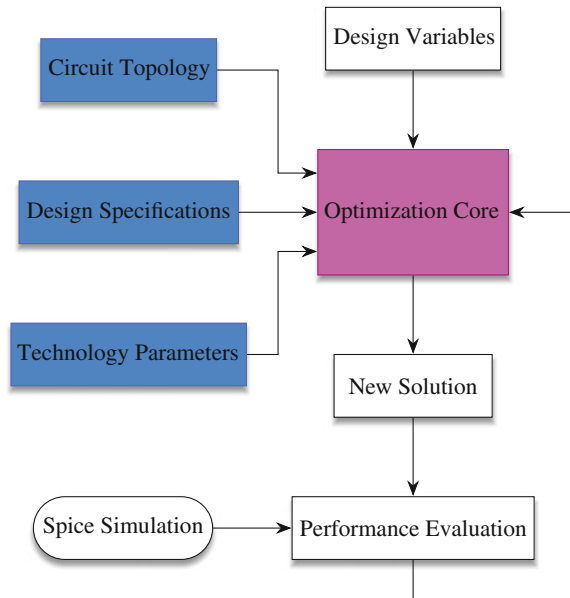


Figure 2.1 shows the execution scheme of an analog optimization-based tool with simulation-based performance evaluation. The tool takes as input the circuit topology, design specifications, and technology parameters. The optimization core generates solutions for the optimization problem according to the implemented optimization technique. For each iteration, it is necessary to evaluate the quality of the generated solution. It is quantified by a cost function, which gives an indication of the performance of the generated solution with respect to the desired specifications. The performance is estimated by SPICE simulation of a set of test benches in which design specifications can be extracted.

2.2.1 Sources of Process Parameter Variability

Submicrometer integrated circuit technologies present high incidence of variability in the fabrication process. These variations affect the performance of ICs, both analog and digital. In digital circuits, the effects are directly perceived in the propagation time of the digital signal. In analog circuits, process variations affect the operation point of the individual transistors and cause mismatch that can lead to loosing circuit functionality [10].

According to Orshansky et al. [27], the variations in the IC fabrication process can be classified in three categories: front end, back end, and variations caused by the environment. Front-end variations are caused by the first steps of device fabrication, such as ion implantation, oxidation, polysilicon deposition, and others.

In these stages, random variations occur in the transistor gate sizes (W and L), threshold voltage (V_{TH}), silicon oxide thickness (T_{ox}), among others.

Back-end variations are characterized by the process variations caused in the metallization step of the fabrication process. The width of metal lines and interconnection vias, at different metal levels, are affected by random variations. At the same time, passive devices—such as capacitors and inductors—present variations around the nominal values, because they are implemented, in general, by metal lines.

The environment variations refer to the differences between the nominal and real operation conditions of the IC. We can cite temperature and supply voltage variations as example. These variations are systematic and can be treated at design level in order to attenuate their effects.

Figure 2.2 illustrates three of the main parameter variation sources present in an IC fabrication process. A random fluctuation of dopants occurs due to the difficulty in controlling the exact quantity and energy of the ion implantation in small devices. Some ions are located at undesired regions, and the concentration presents nonuniform patterns. At the same time, there is a random variation in the effective channel length and width, making it slightly different from the drawn dimensions. Devices with large gate length are less sensitive to process imperfections. Polysilicon width does not produce large variations in W , since this dimension is defined by the diffusion region, which, in general, has a large area [10]. Finally, the gate oxide thickness T_{ox} presents a random variation gradient along the wafer area.

According to Orshansky et al. [27], front-end variations are very relevant for an analog design. In order to exemplify the influence of a variation in the fabrication process over an integrated circuit, consider the threshold voltage V_{TH} . For large channel transistors with uniform doping [34], it can be estimated by

$$V_{TH} = V_{TH0} + \gamma(\sqrt{2|\phi_F| + V_{SB}} - \sqrt{2|\phi_F|}),$$

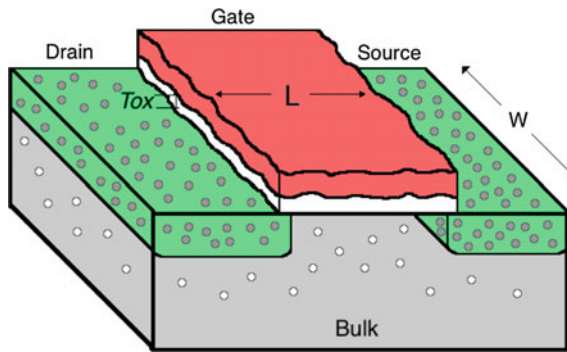


Fig. 2.2 Representation of the main variation sources in the integrated circuit fabrication process: random doping fluctuation, effective gate dimensions, and gate oxide roughness

where V_{TH0} is the threshold voltage for a long-channel device with source-bulk voltage equal to zero, ϕ_F is the Fermi level, and γ is obtained by

$$\gamma = \frac{\sqrt{2q\epsilon_{Si}N_{sub}}}{C_{ox}}.$$

Here, q is the fundamental charge of an electron, ϵ_{Si} is the silicon permittivity, N_{sub} is the density of electrons in the substrate, and C_{ox} is the silicon oxide capacitance.

In this context, we can verify that a variation in the number of dopants in the substrate (N_{sub}) has a great effect in the threshold voltage. The same occurs with a variation in C_{ox} , which is dependent on the gate oxide thickness T_{ox} :

$$C_{ox} = \frac{\epsilon_{ox}}{T_{ox}}.$$

Although well controlled, the gate oxide thickness can present relevant variations over different regions of the wafer.

The standard deviation of V_{TH} can be estimated by Orshansky et al. [27] as follows:

$$\sigma_{V_{TH}} = 3.19 \cdot 10^{-8} \frac{T_{ox} N_{sub}^{0.4}}{\sqrt{L_{eff} W_{eff}}}.$$

We can verify that, besides T_{ox} and N_{sub} , the variation of V_{TH} is related to the inverse square root of the effective gate area. Small devices are, therefore, more sensitive to process variations.

2.2.2 Estimating Circuit Yield

Yield is the ratio of accepted ICs over the total number of fabricated ICs. If the yield is significantly less than 100 %, this implies a financial loss to the IC manufacturer. Therefore, it is important to calculate and maximize the manufacturing yield already during the design stage [12]. This is called design for manufacturability, which implies techniques for yield estimation and yield optimization.

Failures that show up when the IC is in use in a product in the field are even more expensive, for instance when the IC is used under extreme operating conditions such as high temperatures. To try to avoid this, the design has to be made as robust as possible. This is called design for robustness or design for quality, which implies techniques for variability minimization and design centering.

Some works have been done in the analog automatic synthesis theme considering design for manufacturability. The strategy for calculating the yield-aware specification Pareto front is explored by Mueller-Gritschneider and Graeb [25].

Other approaches propose the use of simplified sampling techniques for yield estimation in order to reduce computational time [6, 13, 22]. The use of response surface method (RSM) for circuit performance modeled as quadratic functions of the process parameters is also reported [31]. All of them, however, face challenges in accuracy, compromising the search for the optimized circuit.

The problem is how to estimate yield with accuracy in a reasonable processing time. Monte Carlo is the standard technique for statistical simulation of circuits and for yield estimation during the design phase. The reason for this is that Monte Carlo is applicable to arbitrary circuits, arbitrary statistical models, and all performance metrics of interest, while allowing arbitrary accuracy. On the other hand, circuit size, nonlinearity, simulation time, and required accuracy often conspire to make Monte Carlo analysis expensive and slow. A single Monte Carlo run can cost a few thousand SPICE simulations, and higher accuracy requirements demand longer runs.

A detailed analysis of traditional pseudo-random Monte Carlo sampling, Latin hypercube sampling (LHS), and quasi-Monte Carlo (QMC) techniques is done by Singhee and Rutenbar [32]. The goal is to reduce the number of sample points while keeping the accuracy of the yield prediction. For high-dimensional problems, QMC presents advantages in terms of simulation speed, giving $2\times$ to $8\times$ speedup over conventional Monte Carlo for roughly 1 % accuracy levels.

2.3 Problem Formulation

The problem of analog integrated circuit sizing is modeled as an optimization problem by translating the circuit performance specifications to a cost function dependent on the transistor dimensions, capacitances, resistances, and bias voltages and currents (design free parameters). This cost function fully defines the performance space, which can be explored by an optimization heuristic for a minimum (or maximum) point. The optimized solution is contained in this point.

Consider a set of circuit performance functions (design specifications) $\mathbf{X}(\mathbf{p}, \mathbf{q}) = \{S_1, S_2, \dots, S_k\}$, which depends on a set of design parameter values \mathbf{p} and on a set of technology parameter values \mathbf{q} . S is an individual specification, and k is the number of design specifications. Performance functions for an operational amplifier can be the low-voltage gain (A_{v0}), gain–bandwidth product (GBW), slew rate (SR), dissipated power (P_{diss}), etc. Design parameters are the free variables the designer can handle in order to design the circuit, such as gate dimensions (length L and width W), reference currents, and capacitor values. Technology parameters include electrical MOS model parameters (such as oxide thickness T_{ox} and threshold voltage of the long-channel device at zero substrate bias V_{TH0}), supply voltages, and operation temperature range.

The acceptance of a circuit by a specification test can be expressed as follows:

$$\mathbf{X}(\mathbf{p}, \mathbf{q}) \in \Phi.$$

Φ is the region of acceptable performance specifications in the performance space. The acceptance region Ψ in the design parameter space is defined by

$$\mathbf{X}(\mathbf{p}, \mathbf{q}) \in \Phi \rightarrow \mathbf{p} \in \Psi.$$

A manufactured circuit will be considered acceptable if all of its actual performances fall within acceptable limits, i.e., if $S_i^L \leq S_i \leq S_i^U$, where the indexes L and U correspond to lower and upper specification limits, respectively.

The parameter values \mathbf{q} vary statistically around a nominal value, caused by unavoidable process fluctuations in manufacturing, with a joint probability density function (JPDF) $g(\mathbf{p}, \mathbf{q})$.

Two different types of parameter variation are present in a semiconductor fabrication process: global (interdie) and local (intradie) variation. Global variation of the electrical parameters is induced by process fluctuations in manufacturing, which affect all devices in a circuit in the same way. It is independent of length L and width W .

Local variation induces differences between identically designed devices caused by edge roughness, doping variation, boundary effects, etc. In this case, the variation of L depends on the width of the device. Other parameters such as sheet resistance, channel dopant concentration, mobility, and gate oxide thickness are inversely dependent on the gate area ($W \cdot L$), since the parameters average over a greater distance or area [10]. Mismatch is dominated by local variation and affects the electrical behavior of input differential pairs and current mirrors, even for well-designed layouts.

The manufacturing yield Y of a circuit can be formulated by the number of accepted circuits that pass the specification test over the total number of considered circuits:

$$Y = \text{Prob}(\mathbf{X}(\mathbf{p}, \mathbf{q}) \in \Phi).$$

The manufacturing yield can be estimated by repeating circuit electrical simulation and performance specification evaluation. This is done by Monte Carlo analysis, which simulates the variation of the electrical parameters that affect all devices in a circuit. In order to simulate global variation, the process parameters are randomly selected in each simulation run and globally assigned to all device instances in a design. For local variation simulation, every instance of a device that contains matching-relevant parameters receives an individual random value around a typical mean.

There are some design techniques typically used to improve yield in analog circuits [7]. These techniques can be implemented—in combination or not—in three design stages: topology selection, transistor sizing, and physical synthesis

[24]. The focus of this chapter is restricted to yield maximization at the transistor sizing stage.

In our approach, yield maximization is performed by determining a set of nominal values of the design parameters, \mathbf{p} , that maximizes the probability of the random performances lying within Φ [4]. However, there is not an explicit for estimating $g(\mathbf{p}, \mathbf{q})$. The yield optimization problem can be formulated in the space of independent statistical disturbances. The region of acceptability in the disturbance space contains all possible combinations of disturbances that can occur in the manufacturing of a circuit, which, for specific nominal parameter values, do not result in unacceptable performance. Yield optimization is, therefore, performed by modifying the acceptability region in a way that increases the coverage of a fixed probability distribution [9].

This yield can be calculated in both the device parameter space and the circuit performance space. This calculation, however, is complicated by the fact that, in either space, one of the two elements is not known explicitly: the statistical fluctuations are known in the device parameter space but not in the circuit performance space, whereas the acceptability region is known in the performance space but not in the parameter space [12]. Monte Carlo simulation, combined with an optimization procedure, is the most effective way to estimate acceptability region in the performance space.

2.3.1 Transistor Sizing as an Optimization Problem

The proposed approach for optimization of circuit performance explores the yield prediction as a design objective in an automatic sizing procedure. It is a nonlinear programming problem and requires the formulation of a single performance function (cost) to minimize subject to a set of inequality constraints, as in the following standard form [26]:

$$\begin{aligned} & \underset{\mathbf{p}, \mathbf{q}}{\text{minimize}} && F_m(\mathbf{p}, \mathbf{q}), \quad i = 1, \dots, M \\ & \text{subject to} && C_n(\mathbf{p}, \mathbf{q}) \leq C_{n(\text{ref})}, \quad n = 1, \dots, N \end{aligned}$$

where M is the total number of F_m specifications to optimize, and N is the number of C_n constrained performance functions. Here, \mathbf{X} can be rewritten as a set of design objectives and design constraints:

$$\mathbf{X}(\mathbf{p}, \mathbf{q}) = \{F_1, \dots, F_M, C_1, \dots, C_N\}.$$

$C_n(\mathbf{p}, \mathbf{q})$ is a function that is dependent on the specification type: minimum required value ($C_{\min}(\mathbf{p}, \mathbf{q})$) or maximum required value ($C_{\max}(\mathbf{p}, \mathbf{q})$) [5].

These functions are shown in Fig. 2.3, where a is the maximum or minimum required value, and b is the bound value between acceptable and unacceptable performance values.

Acceptable but non-feasible performance values are the points between a and b . They return intermediate values for the constraint functions in order to allow the exploration of disconnected feasible design space regions. These functions return additional cost for the cost function if the performance is outside the desired range. Otherwise, the additional cost is zero.

The constrained problem can be transformed into an unconstrained minimization problem using the penalty function approach:

$$f_c(\mathbf{p}, \mathbf{q}) = \sum_{m=1}^M w_m \cdot \hat{F}_m(\mathbf{p}, \mathbf{q}) + \sum_{n=1}^N v_n \cdot \hat{C}_n(\mathbf{p}, \mathbf{q}). \quad (2.1)$$

Here, w_m and v_n are weights that indicate the relative importance of design objectives and design constraints, respectively. \hat{F} and \hat{C} are normalized design objective and design constraint functions, in order to keep all sum factors in the same order of magnitude.

Yield prediction can be easily included as a design objective in the cost function by adding a new term $\hat{Y}(\mathbf{p}, \mathbf{q}, \epsilon_q)$ in the penalty function:

$$f_c^Y(\mathbf{p}, \mathbf{q}, \epsilon_q) = \sum_{m=1}^M w_m \cdot \hat{F}_m(\mathbf{p}, \mathbf{q}) + \sum_{n=1}^N v_n \cdot \hat{C}_n(\mathbf{p}, \mathbf{q}) + \hat{Y}(\mathbf{p}, \mathbf{q}, \epsilon_q). \quad (2.2)$$

This new term is dependent on the variability vector of technology parameters \mathbf{q} given by ϵ_q .

We define the characteristic function of Φ as

$$I_{\Phi}(\mathbf{X}) = \begin{cases} 1 & \text{if } \mathbf{X} \in \Phi \\ 0 & \text{if } \mathbf{X} \notin \Phi \end{cases}$$

which is 1 for pass and 0 for fail. This is also known as the indicator function.

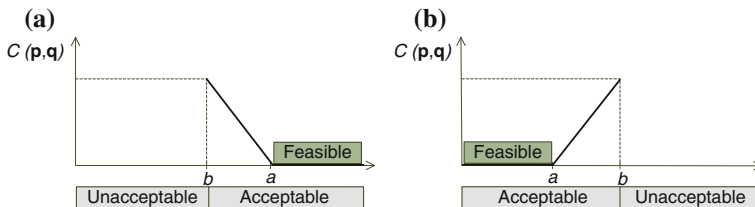


Fig. 2.3 Design constraint performance metrics: **a** minimum required value specifications and **b** maximum required value specifications

We can now define the circuit yield as the probability of a circuit instance lying in the acceptance region:

$$Y = \int_{\mathfrak{R}^*} I_{\Phi}(\mathbf{X})g(\mathbf{X})d\mathbf{X}.$$

There are other ways of calculating the yield, such as the use of process capability indexes (Cpk). The key idea is normalizing the distance to the feasible boundary by the standard deviation (σ) of its performance distribution to consider different process sensitivity. The application of these indexes, however, is out of the scope of this work.

In this work, the cost function is estimated by electrical simulations performed by Synopsys HSpice[®]. The high computational cost of Monte Carlo simulations is diminished by performing circuit statistical analysis only for a subset of design solutions in the optimization process, as detailed in the next section.

2.3.2 Monte Carlo in the Optimization Flow

According to Eq. 2.2, we need to estimate circuit yield for the calculation of the cost function in the optimization flow. However, it is computationally costly if done by Monte Carlo simulation. On the other hand, if yield prediction is not considered, the optimization algorithm tends to find optimal solutions close to the border of the performance space. At these points, a small variation in the process parameters makes the performance specifications fall outside the acceptable region. Thus, a strategy for dealing with this problem must be included in the optimization process. It is called design centering.

There are two approaches for improving processing time considering Monte Carlo simulations: to reduce the number of iterations in which Monte Carlo simulations are necessary; and to reduce the number of runs in a Monte Carlo simulation. Both strategies are essential for improving the processing time during the search for an optimal design solution.

2.3.2.1 Reducing the Number of Monte Carlos Simulations

It is possible to reduce the number of iterations in which the calculation of Y is necessary by analyzing the influence of this term over the entire cost function of Eq. 2.2. Figure 2.4 illustrates this strategy. Consider first the cost function in Eq. 2.1, which does not include yield. If the current solution is not a best solution candidate, i.e., if it is already a worst solution even without the calculation of Y , this solution can be discarded and Monte Carlo simulation is not necessary. As $f_c^Y(\mathbf{X})$ is unknown before the Monte Carlo simulation, the test for best solution candidate is

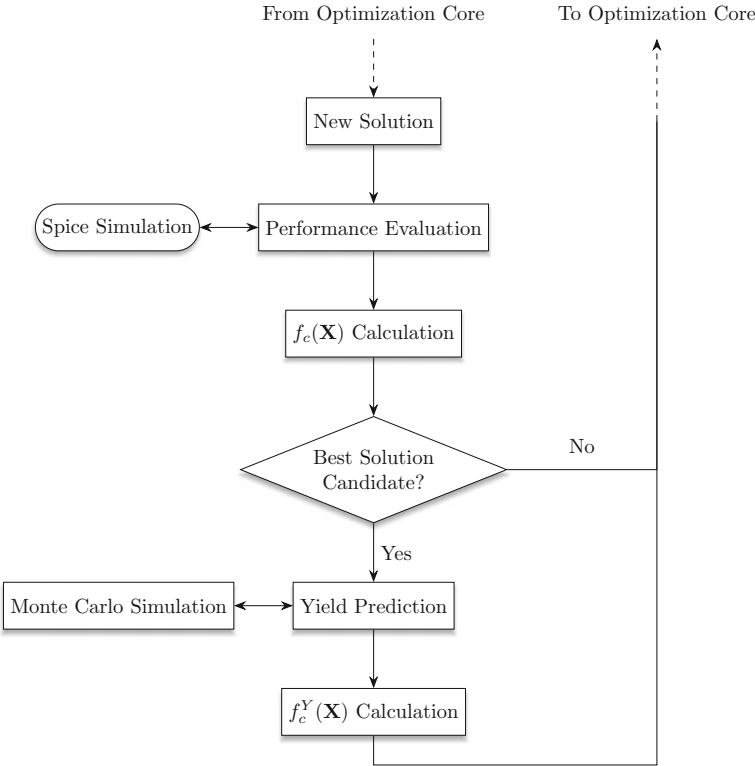


Fig. 2.4 Optimization flow including yield prediction only for best solution candidates

done by the calculation of $f_c(\mathbf{X})$. Monte Carlo simulation is executed only when $f_c(\mathbf{X}) < \min(f_c^Y(\mathbf{X}))$ —where $\min(f_c^Y(\mathbf{X}))$ is the current best solution.

At the start of the optimization process, the current best solution is frequently updated. However, the update frequency tends to reduce along iterations. Consequently, the number of iterations in which Monte Carlo simulation is necessary also reduces. Computational time is spared, since useless Monte Carlo simulations are avoided.

2.3.2.2 Reducing the Number of Runs in a Monte Carlo Simulation

The reduction in the number of runs in a Monte Carlo simulation can be implemented with the calculation of the number of samples (n) necessary to achieve a desired confidence level in the yield estimation.

The expected value of a random variable p is $\mu = E(p)$. If we generate values p_1, \dots, p_n independently and randomly from the distribution p , we can estimate μ as

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n p_i.$$

One of the great strengths of the Monte Carlo method is that the sample values themselves can be used to get a rough idea of the error $\hat{\mu}_n - \mu$. The average squared error in Monte Carlo sampling is σ^2/n . The most commonly used estimation of standard deviation σ^2 is

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (p_i - \hat{\mu}_n)^2.$$

Monte Carlo sampling typically uses large values of n for guaranteeing that this estimation is a good approximation to the actual σ^2 .

A variance estimate σ^2 tells us that the error is on the order of σ/\sqrt{n} . We know that $\hat{\mu}_n$ has mean μ and we can estimate its variance by σ^2/n .

From the central limit theorem (CLT), we also know that $\hat{\mu}_n - \mu$ has approximately a normal distribution with mean 0 and variance σ^2/n . The CLT can be used to get approximate confidence intervals for μ . For 95 % confidence interval,

$$\mu_{95\%} = \hat{\mu}_n \pm 1.96 \frac{\sigma}{\sqrt{n}}.$$

For 99 % confidence interval,

$$\mu_{99\%} = \hat{\mu}_n \pm 2.58 \frac{\sigma}{\sqrt{n}}. \quad (2.3)$$

In a general way,

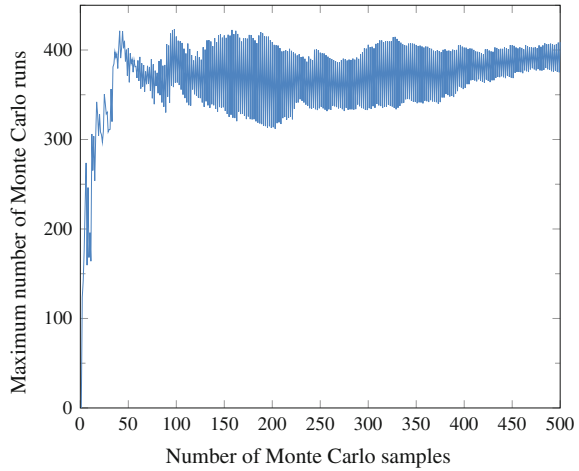
$$\mu_{c\%} = \hat{\mu}_n \pm \Phi^{-1}(1 - \alpha/2) \frac{\sigma}{\sqrt{n}},$$

where $\alpha = 1 - c/100$ and $\Phi(\cdot)^{-1}$ is the inverse cumulative distribution function (ICDF) of $N(0, 1)$, the standard normal distribution. It is not available in closed form, and computation requires careful use of numerical procedures. It is also called “probit function,” an acronym for “probability unit function.” The probit function can be calculated as

$$\text{probit}(p) = \sqrt{2} \cdot \text{erf}^{-1}(2p - 1).$$

In Matlab, the `erfinv` function is available for erf^{-1} (inverse error function).

Fig. 2.5 Example of the maximum number of Monte Carlo runs calculated with Eq. 2.3. The circuit is a Miller amplifier, and the specification being measured is low-voltage gain. A good choice for n_0 is 50, the value where the graph stabilizes around 400



The steps for determining the number of Monte Carlo runs that matches a 99 % confidence interval for a determined specification are the following:

```

Initialize  $n_0$ ;
 $i = 0$ ;
repeat
     $i = i + 1$ ;
    Run Monte Carlo simulation;
    Calculate  $\hat{\sigma}_i$  and  $\Delta_\mu$ ;
     $n_i = (\frac{\hat{\sigma}_i}{\Delta_\mu} \cdot 2.58)^2$ ;
until  $n_i > n_{i-1}$ ;

```

The correct choice of n_0 is fundamental for correctly estimating n_{i+1} , since the confidence of σ is dependent on n . With some simulations, we can infer that $n_0 = 50$ is a good choice. Figure 2.5 shows the simulation of Eq. 2.3 for different values of n for estimating the low-voltage gain of a Miller OTA. One can note that the graph stabilizes for $n = 50$, indicating a maximum number of samples (n_{i+1}) equal to 400.

2.4 The UCAF Tool

UCAF is a CAD tool we developed for the automatic design of analog basic blocks including yield optimization. The tool sizes an integrated circuit by modeling it as an optimization problem and exploring efficiently the design space searching for optimal solutions. The main design flow of this tool is the same shown in Fig. 2.1. This general design flow is implemented in Matlab, and Synopsys HSpice is used for performance estimation.

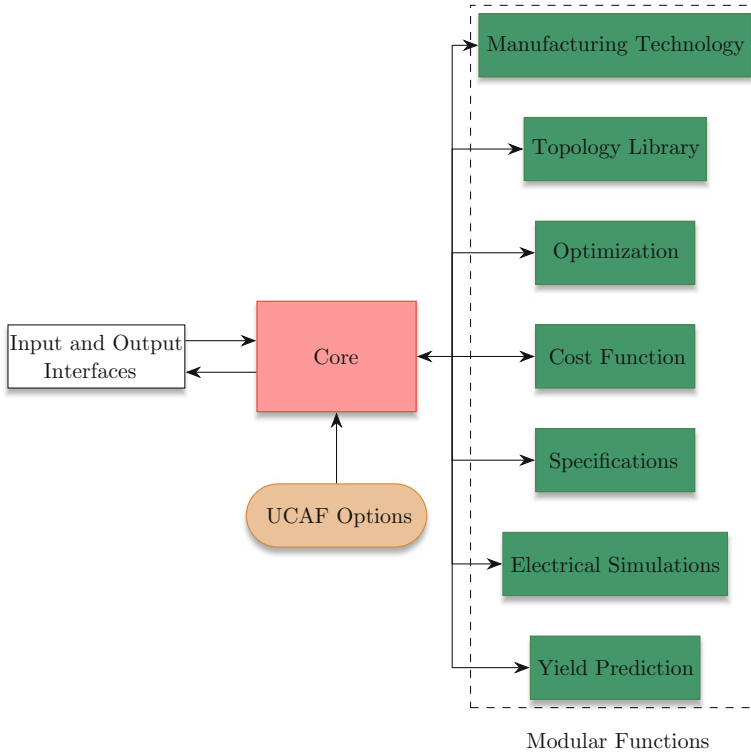


Fig. 2.6 UCAF structure

The implementation of UCAF is made using modular functions to solve a generic analog IC design. These modular functions are shown in Fig. 2.6. Modularity allows a high degree of configurability, since each function can be substituted by a similar one without losing functionality. For example, the optimization algorithm can be changed independently of the remaining functions. The tool is configured with the aid of a specific script as the input interface. Also, a graphical interface guides the user through the basic configurations. The output interface presents the generated solutions.

The “Core” module is the main function, which creates and organizes a new design, creates design folders, sets the modular functions, writes the simulation file, and performs other important tasks.

The function “Manufacturing Technology” implements the interface between the design and the fabrication technology. It reads and configures the parameters of simulation models from the design kit provided by the foundry.

Each new analog block inserted in the tool is saved in a cell library. This task is performed by the “Topology Library” function. These cells can be reused for different design specifications.

The “Optimization” function is responsible for the optimization algorithm that will guide the design space exploration. This function will be detailed further in Sect. 2.4.2.

The “Cost Function” implements Eq. 2.1 and is responsible for representing the design as an unconstrained minimization problem. In order to evaluate the cost function, it is necessary to estimate the values of the circuit specifications. Thus, the UCAF tool has the “Electrical Simulation” and “Specifications” functions. These functions are analyzed with more details in Sect. 2.4.3.

2.4.1 A Simplified Design Example

The design flow of the UCAF tool can be illustrated by the sizing of a simple active load differential amplifier circuit, as shown in Fig. 2.7. It is composed of four transistors and a tail current (I_{ref}). In order to simplify the design, we assume all transistors have the same size and I_{ref} is fixed at 10 μ A. This is not of practical use, but reduces the design problem to only two variables: the transistors channel width (W) and length (L). It also allows the visualization of the design space and provides an intuitive understanding of the optimization procedure.

Fig. 2.7 Active load differential amplifier

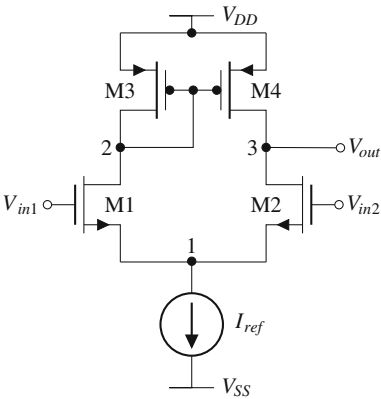


Table 2.1 Required specifications for the differential amplifier of Fig. 2.7

| Specification | Required value |
|--|-----------------|
| Gate area | Minimize |
| Low-frequency gain (A_{vo}) | ≥ 40.00 dB |
| Phase margin (PM) | ≥ 70.00 |
| Gain-bandwidth product (GBW) | ≥ 1.00 MHz |
| Input common-mode range (ICMR ⁺) | ≥ 0.40 V |

Table 2.1 summarizes the circuit performance specifications. The design objective is to minimize the gate area ($W \cdot L$). The cost function is calculated by

$$f_c(\mathbf{X}) = \frac{1}{\text{Area}_{\text{ref}}} \cdot \text{Area}(\mathbf{X}) + \sum_{n=1}^N C(\mathbf{X}), \quad (2.4)$$

where \mathbf{X} is the vector of free variables ($\mathbf{X} = [WL]$), Area_{ref} is the weighting parameter of the gate area, and $C_n(\mathbf{X})$ represents the constraint performance metric for the required specifications. $\sum C(\mathbf{X})$ is calculated by

$$\begin{aligned} \sum C_n(\mathbf{X}) = & C_{\min}(A_v(\mathbf{X}), A_{v\text{ref}}) + C_{\min}(\text{PM}(\mathbf{X}), \text{PM}_{\text{ref}}) \\ & + C_{\min}(\text{GBW}(\mathbf{X}), \text{GBW}_{\text{ref}}) + C_{\min}(\text{ICMR}^+(\mathbf{X}), \text{ICMR}_{\text{ref}}^+), \end{aligned}$$

where C_{\min} and C_{\max} are the constraint performance functions shown in Fig. 2.3.

Assuming W can vary between 0.22 and 10 μm and L between 0.2 and 1 μm in steps of 0.05 μm , the optimization problem has 3120 possible solutions. We exhaustively calculated all 3120 solutions by electrical simulation. The resulting design space is shown in Fig. 2.8, where one can see the high nonlinearity of the

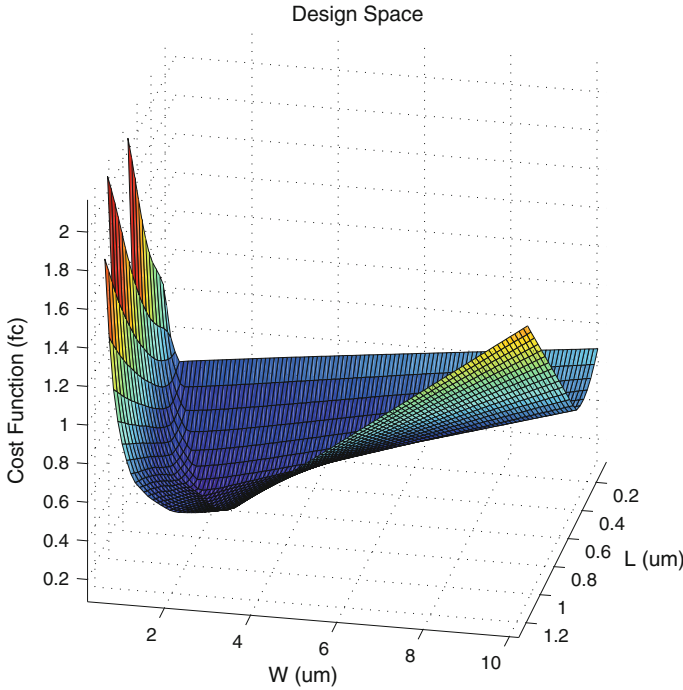


Fig. 2.8 Resulting design space composed by 3120 possible solutions for the differential amplifier of Fig. 2.7

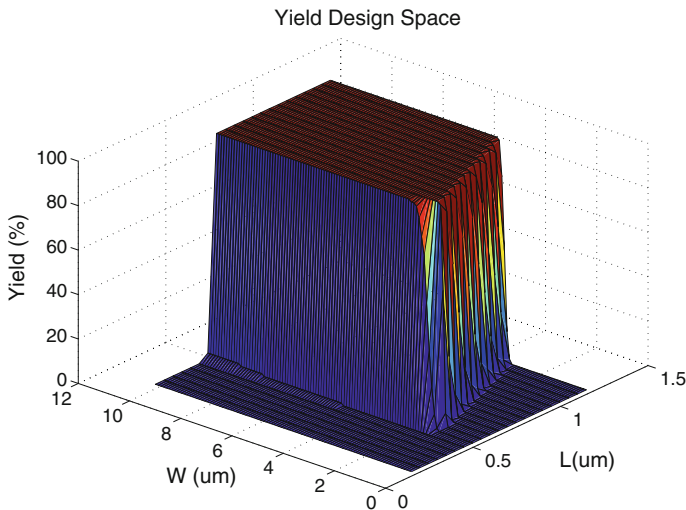


Fig. 2.9 Yield design space estimated by Monte Carlo simulation

cost function with respect to the design free variables. The optimization method searches the design space in order to find the value of W and of L that make the cost function have the lowest value. The optimal solution is reached at the point $W = 1.62 \mu\text{m}$ and $L = 0.54 \mu\text{m}$, with a cost value of 0.11958.

The yield of each solution is evaluated by Monte Carlo simulations with 200 samples. The yield design space is shown in Fig. 2.9. It is possible to see that the design space is abruptly deformed at the region where the yield moves from 100 to 0 %. The optimal solution not considering yield represents, in practice, an yield of 51.8 %, indicating that the solution is located in a performance region very sensitive to process variations.

Using Eq. 2.2, the design spaces of Figs. 2.8 and 2.9 can be joined to result in a design space including yield prediction, shown in Fig. 2.10. The optimal solution is now at $W = 1.82 \mu\text{m}$ and $L = 0.56 \mu\text{m}$, with a cost value of 0.13658 and yield of 99.8 %. The optimal point moved just slightly, but enough for considerably increasing the yield. The difference between this optimal value and the optimal value without yield is $\Delta W = 0.02 \mu\text{m}$ and $\Delta L = 0.02 \mu\text{m}$.

2.4.2 Optimization

This is the main function of the UCAF tool, because it is responsible for exploring the design space. Here we opted for using genetic algorithms (GA), available in the Matlab Genetic Algorithm Optimization Toolbox (GAOT) [17].

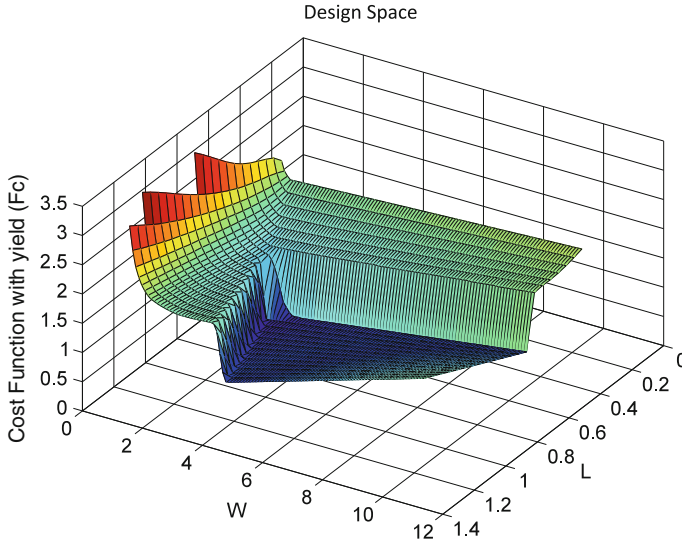


Fig. 2.10 Design space considering yield prediction

The GA optimization approach is based on the biology theories of evolution and genetics. It is a non-deterministic meta-heuristic and can be used for optimizing nonlinear functions [11].

A specialized vocabulary is used for better reflecting the biological approach. A solution is called an “individual,” and since GA work with a number of solutions simultaneously, this set of solutions is called “population.” The iterations of the optimization process are called “generations,” and the cost function is referred as the “fitness” function.

In each generation, the individuals of the current population are crossed, generating new individuals that share characteristics from both parents (“crossover”) and that may suffer “mutation.” Each individual is represented as a chromosome, which in turn represents the optimization variables and their values.

Figure 2.11 shows the flowchart for optimizing the circuit size using GA. The GA core receives three inputs: the configuration parameters, the design specifications, and the technology parameters. The first step is creating an initial set of solutions, which is randomly performed by an initialization function. Each solution is then evaluated according to the fitness (cost) function, given by Eq. 2.4, but replicated here for convenience (recall that circuit specifications are estimated via electric simulations):

$$f_c(\mathbf{X}) = \frac{1}{\text{Area}_{\text{ref}}} \cdot \text{Area}(\mathbf{X}) + \sum_{n=1}^N C(\mathbf{X}).$$

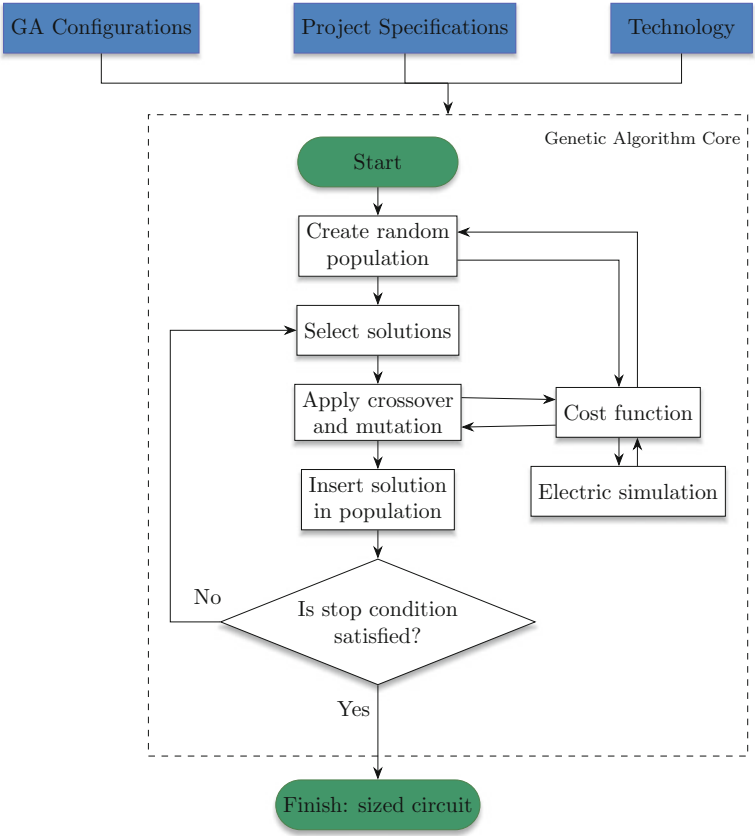


Fig. 2.11 Genetic algorithm flowchart for circuit sizing

Afterward, a subset of the solutions is chosen based on a selection function. The GAOT accepts the following selection functions: roulette wheel, normalized geometric rank selection, and tournament selection. The roulette wheel function first gives a normalized probability for each solution based on its fitness and then builds a roulette based on these probabilities. A random number is generated, and the solution with this number is selected. That way, the better the fitness, the higher the chance of being selected.

The selection by ranking orders the solutions based on their fitness and assigns a selection probability to each position. As with the roulette function, a solution is randomly chosen according to the probabilities. The difference here is that the probability of selection of a solution does not directly depend on its fitness, but only on its rank.

The third selection function chooses a number of solutions uniformly at random and keeps only the best solution. New tournaments are drawn, and the best overall solution is kept.

After selecting a subset of solutions, the next step in Fig. 2.11 is to perform crossover and mutation over the solutions. These operations are responsible for the state space search process of GA, since they generate new solutions. The crossover operation takes two solutions (chromosomes), splits them at some random point, and then combines one part from each chromosome, generating two new chromosomes. The mutate operation takes a single chromosome, selects a random point, and then inverts its value. Various functions for crossover and mutation are described in Houck et al. [16].

The new population is tested and, if the stop condition is satisfied, the circuit is sized and the process finishes. If it is not satisfied, a new generation (iteration) is performed. A maximum number of generations or a minimum cost function difference can be used as stop conditions.

2.4.3 Circuit Characterization

The optimization procedure has an interface with an external electrical simulator to estimate the circuit performance. For each specification, it is necessary to simulate a circuit test bench, performing AC, DC, or transient analysis. The current version of UCAF tool has some circuit standard test benches to measure the specifications of operational amplifiers [2, 30]. These test benches are shown in Fig. 2.12.

An AC analysis is performed for measuring the low-frequency gain (A_{v0}), the gain–bandwidth product (GBW), and the phase margin (PM). The configuration is shown in Fig. 2.12a. The results of this simulation can be plotted as a Bode diagram. From the gain curve, A_{v0} and GBW specifications are extracted. In the same way, the phase margin is obtained in the phase curve, as shown in Fig. 2.13. In UCAF, this extraction is performed by the “Specification” modular function.

To obtain the input common-mode range (ICMR), the amplifier is connected in unity gain configuration, as shown in Fig. 2.12b. In this simulation, the input voltage is varied from a minimum to a maximum level through a DC analysis. Positive and negative values are obtained from simulation output when the gain is linear.

Figure 2.12c shows a circuit with a voltage gain of -10 . This circuit is used for measuring the output swing (OS) with a DC analysis of input voltage sweep. As the gain is -10 , the output level of saturation is obtained. The difference between the minimum and maximum output levels is the OS specification.

The response speed of an amplifier (Slew Rate) is measured with the same configuration as ICMR. The goal of this simulation is the transient analysis of a step response of the circuit through the verification of the raise or fall behavior of the output voltage level, as illustrated in Fig. 2.12d.

The common-mode rejection ratio (CMRR) is given by the ratio of the common voltage (V_{cm}) by the generated output voltage. This specification represents the rejection amount of the input common-mode voltage due to the non-idealities of the amplifier. To measure this specification, an AC analysis is executed using the

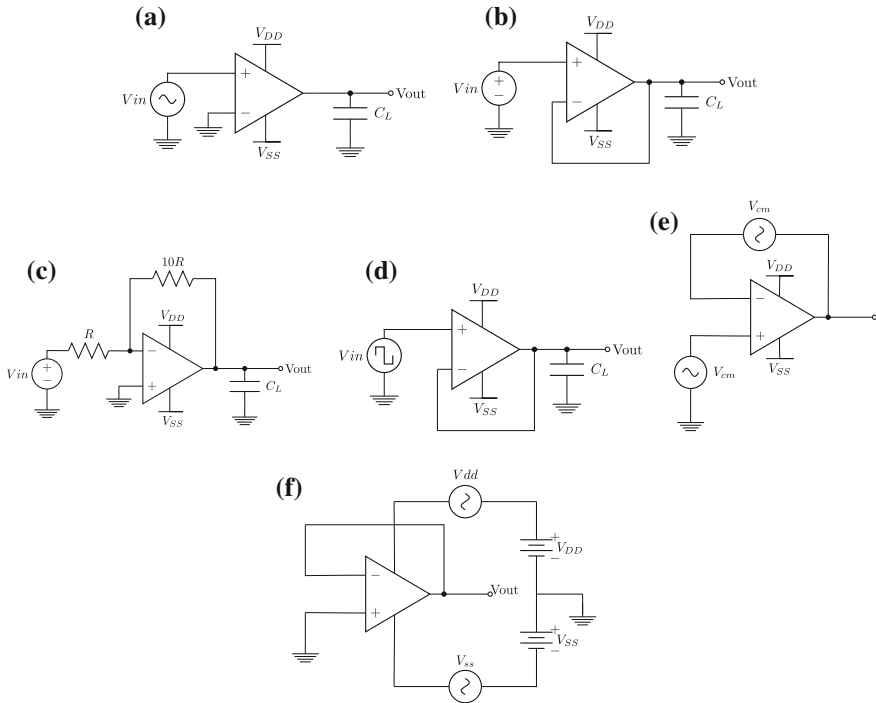


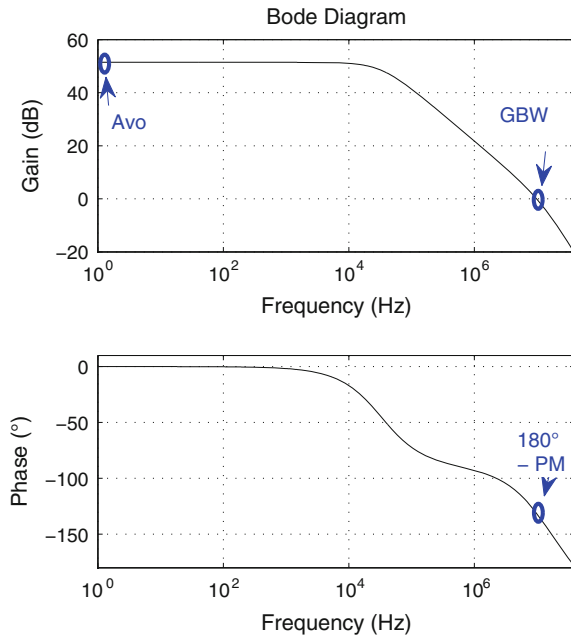
Fig. 2.12 Implemented test benches for measuring the performance of operational amplifiers. **a** AC open loop. **b** ICMR. **c** Output swing. **d** Slew rate. **e** CMRR. **f** PSRR

configuration shown in Fig. 2.12e, varying the operation frequency of the common-mode voltage source.

Like the CMRR, the power supply rejection ratio (PSRR) indicates the amplifier rejection capacity with respect to the noise coming from the power supply of the circuit. The circuit used to measure the PSRR is shown in Fig. 2.12f. The noise comes from the V_{DD} and from the V_{SS} power supplies, resulting in positive (PSRR⁺) and negative (PSRR⁻) rejection ratio, respectively. An AC analysis is executed to sweep the frequency of the voltage sources, simulating the noise coming from the power supplies. It is important to notice that these two simulations are performed separately.

With a multi-core computer architecture, the electrical simulation task can be carried out in parallel in different cores, since each specification has an independent test bench. The UCAF implementation is capable of using all cores simultaneously to simulate the circuit, resulting in a relevant reduction in the overall processing time.

Fig. 2.13 Bode diagram used to extract low-frequency gain (A_{v0}), gain–bandwidth product (GBW), and phase margin (PM)



2.5 Automatic Design of a Two-Stage Amplifier

To illustrate the application of the optimization procedure described above, we performed the automatic design of a two-stage CMOS Miller operational transconductance amplifier (OTA). The schematics of this amplifier is shown in Fig. 2.14. It is composed of an input differential amplifier as first amplification stage

Fig. 2.14 Schematics of a two-stage Miller amplifier

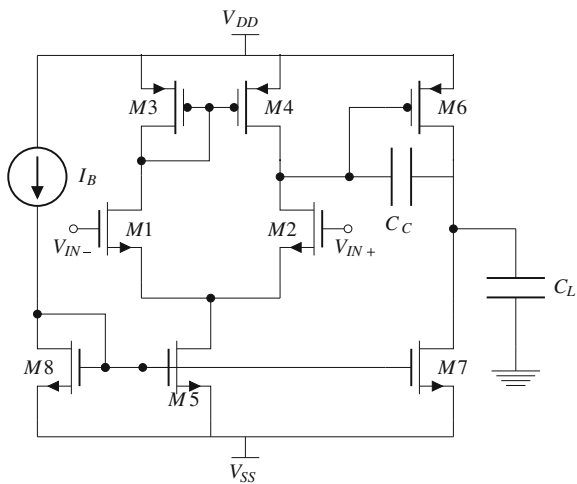


Table 2.2 Upper and lower bounds for the free values in the design of a Miller OTA

| Variable | Lower bound | Upper bound |
|----------|--------------------|----------------------|
| W_i | 0.22 μm | 50.00 μm |
| L_i | 0.18 μm | 10.00 μm |
| I_B | 0.10 μA | 100.00 μA |
| C_c | 0.10 pF | 10.00 pF |

and by an inverter amplifier in the second stage. A compensation capacitor (C_c) is connected between stages for stability purposes [2].

This design has the following user specifications as input constraints: low-frequency gain (A_{vo}), gain–bandwidth product (GBW), phase margin (PM), slew rate (SR), common-mode input range (ICMR), and output swing (OS). The current mirrors and the differential pair result in the following matched transistors: $M_1 = M_2$, $M_3 = M_4$ and $M_5 = M_8$. Designing this circuit requires calculating all CMOS gate sizes (W and L), the bias current source (I_B), and the compensation capacitor (C_c), resulting in the following 12 free variables: W_1 , L_1 , W_3 , L_3 , W_5 , L_5 , W_6 , L_6 , W_7 , L_7 , I_B , and C_c .

The target fabrication technology is XFAB 0.18 μm , which defines the minimal values of transistor sizes: $L_{\min} = 0.18 \mu\text{m}$ and $W_{\min} = 0.22 \mu\text{m}$, with a grid (λ) of 0.01 μm . The variable bounds are shown in Table 2.2. The design space has 12 dimensions and 2.76×10^{40} possible solutions, and thus, cannot be exhaustively explored with current ordinary computational resources.

Table 2.3 Design specifications and results for the Miller amplifier

| Specification | Required value | Automatic design 1 | Automatic design 2 | Automatic design 3 | Jafari et al. [19] | Liu et al. [21] |
|-------------------------------------|----------------|--------------------|--------------------|--------------------|--------------------|-----------------|
| A_{vo} (dB) | ≥ 70.00 | 73.55 | 77.25 | 76.17 | 82.40 | 80.66 |
| GBW (MHz) | ≥ 2.00 | 2.32 | 2.39 | 4.17 | 9.77 | 2.04 |
| PM ($^\circ$) | ≥ 50.00 | 55.69 | 54.17 | 66.58 | 60.00 | 55.60 |
| SR (V/ μs) | ≥ 5.00 | 5.19 | 8.17 | 6.01 | 5.07 | 1.50 |
| ICMR ⁺ (V) | ≥ 0.60 | 0.76 | 0.83 | 0.82 | – | – |
| ICMR [–] (V) | ≤ -0.60 | -0.72 | -0.68 | -0.64 | – | – |
| OS (V) | ≥ 1.00 | 1.17 | 1.65 | 1.61 | 1.17 | 1.91 |
| Power dissipation (μW) | Min | 25.82 | 108.49 | 144.16 | 52.00 | 1114.40 |
| Gate area (μm^2) | Min | 16.88 | 299.61 | 589.60 | 236.25 | 1407.78 |
| Yield (%) | Max | 25.62 | 92.11 | 100 | – | – |
| Execution | – | 43.55 | 239.05 | 139.58 | – | 164.42 |
| Time (min) | | | | | | |

The UCAF tool was set to use genetic algorithms with 100 binary individuals, simple mutation, simple crossover, and roulette wheel selection function. The stop criterion is the execution of 1000 generations, and the initial solution is randomly generated. The cost function has power dissipation and gate area as design objectives, and the remaining user specifications are the constraints. The required values of the specifications are shown in the second column of Table 2.3.

The tool was executed in an Intel i7 processor with 8 GB of main memory. Three different configurations were executed. The performance results are shown in the third to fifth columns of Table 2.3, and device sizes are presented in Table 2.4.

The Automatic Design 1 was performed by UCAF without yield analysis in the optimization procedure. This execution spent 43.55 min, and the final result satisfied all constraints. The resulting power and area are equal to 25.82 μW and 16.88 μm^2 , respectively. Comparing with the designs presented by Jafari et al. [19] and Liu et al. [21] for the same circuit, the values obtained by UCAF achieve a considerable reduction in area and dissipated power.

To estimate yield of Automatic Design 1, we executed a Monte Carlo simulation with 2000 samples. The resultant yield was 25.62 %, which is a very low productivity index since approximately 3 of 4 of the fabricated circuits will not satisfy the required specifications. This low yield was expected, because we did not consider it in the optimization problem. The generated solution is very close to the border of the performance space and is very sensitive to random fabrication process variations in this region. Even a small variation causes the solution to move out of the acceptable performance space. For example, the slew rate specification is proportional to the bias current I_B :

$$\text{SR} = \frac{I_B}{C_c}.$$

At the same time, power dissipation depends on I_B :

$$P_{\text{diss}} = (V_{\text{DD}} - V_{\text{SS}}) \cdot (2 \cdot I_B + I_7).$$

The optimization algorithm tries to satisfy the minimum SR constraint with the smallest possible I_B . This implies in reducing SR to the minimum value, moving the

Table 2.4 Generated solutions for the Miller amplifier designs

| Variable | Automatic design 1 | Automatic design 2 | Automatic design 3 |
|---|--------------------|--------------------|--------------------|
| W_1/L_1 ($\mu\text{m}/\mu\text{m}$) | 2.87/0.27 | 2.89/5.87 | 38.99/0.58 |
| W_3/L_3 ($\mu\text{m}/\mu\text{m}$) | 5.84/0.21 | 27.61/1.95 | 35.97/2.80 |
| W_5/L_5 ($\mu\text{m}/\mu\text{m}$) | 1.42/0.90 | 8.99/5.39 | 10.68/0.86 |
| W_6/L_6 ($\mu\text{m}/\mu\text{m}$) | 31.90/0.26 | 43.87/0.52 | 31.58/4.55 |
| W_7/L_7 ($\mu\text{m}/\mu\text{m}$) | 2.36/0.82 | 6.06/6.34 | 21.12/8.6 |
| I_B (μA) | 2.05 | 35.02 | 35.73 |
| C_c (pF) | 1.00 | 1.26 | 6.29 |

solution point close to the border of the performance space. When not considering yield prediction during the optimization procedure, all analog IC automatic design tools tend to generate solutions with very low yield.

The other two configurations shown in Table 2.3 include yield prediction in the optimization flow. The Automatic Design 2 was performed using MC simulation for yield prediction for best solution candidates. As formulated in Sect. 2.3, this strategy is used to reduce the number of Monte Carlo simulations. The number of MC samples was fixed in 400. The generated solution satisfied all design constraints with 92.11 % of yield. The optimization process consumed 239.05 min, which is 5.5 times slower than the process without considering yield. However, the final yield increased to a practical value.

The Automatic Design 3 was executed with the same previous configuration, but using the central limit theorem with 99 % confidence interval for determining the number of MC samples in each iteration. The resulting solution showed 100 % of predicted yield in 139.58 min of execution. With less processing time, this configuration allowed a better exploration of the design space, leading to a solution point with 100 % of yield.

The relevant increase in circuit yield in the Automatic Designs 2 and 3 is given at the expense of gate area and power dissipation. This agrees with the Pelgrom's law [28], since the variation in the circuit performance is inversely proportional to the square root of the gate area. Another characteristic of the yield optimization results is that the distances between constraint values and reached specifications are increased. This guarantees that specifications fall inside the performance space even concerning the variations in the fabrication process.

2.6 Conclusion

The automatic design of a two-stage Miller amplifier including yield prediction in the optimization flow generated a robust solution with 100 % of yield within a 99 % confidence interval. All performance specifications were met, and gate area and power dissipation were minimized.

The UCAF tool uses genetic algorithms for efficiently searching the design space and produces reliable solutions, since the performance is estimated by means of electrical simulation. The tool executes Monte Carlo simulations for yield prediction, providing a realistic estimation of the performance sensitivity with respect to fabrication process variations. The computational time is reduced by executing Monte Carlo simulation only for best solution candidates and by calculating the minimum number of MC samples for a given theoretical confidence interval.

We implemented an efficient yield-oriented sizing tool that generates robust solutions, contributing for the increase in the number of first-time-right analog integrated circuit designs.

The technique described in this work addresses the optimization of a single basic analog block (a subsystem of a large analog circuit). When maximizing the yield of

the whole circuit, losses might be caused by unmatched interconnections and parasitic effects might appear when integrating subsystems on the top level. Nevertheless, it is necessary to optimize the yield of each subsystem in order to achieve a maximized yield in the whole circuit.

One of the drawbacks of our approach is dealing with large circuits composed by several design variables. The computational cost rapidly increases with the number of design variables, since the design space to be explored grows exponentially. A practical strategy is to divide a large circuit into smaller parts and then size each subcircuit at a time using our proposed methodology. After that, mixed-mode simulation could be considered as an alternative to increase simulation speed for these circuits.

References

1. Ali, S., Wilcock, R., Wilson, P., Brown, A.: Yield model characterization for analog integrated circuit using Pareto-optimal surface. In: 2008 15th IEEE International Conference on Electronics, Circuits and Systems, IEEE, vol. 2, pp. 1163–1166 (2008). doi:[10.1109/ICECS.2008.4675065](https://doi.org/10.1109/ICECS.2008.4675065). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4675065>
2. Allen, P.E., Holberg, D.R.: CMOS Analog Circuit Design, 2nd edn. Oxford University Press, Oxford (2002)
3. Alpaydin, G., Balkir, S., Dundar, G.: An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. IEEE Trans. Evol. Comput. **7**(3), 240–252 (2003). url: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1206446
4. Antreich, K., Koblitz, R.: Design centering by yield prediction. IEEE Trans. Circ. Syst. **29**(2), 88–96 (1982). doi:[10.1109/TCS.1982.1085115](https://doi.org/10.1109/TCS.1982.1085115). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1085115>
5. Barros, M., Guilherme, J., Horta, N.: Analog Circuits and Systems Optimization Based on Evolutionary Computation Techniques. Springer, Berlin (2010)
6. Beer, M., Spanos, P.: Neural network based Monte Carlo simulation of random processes. In: Proceedings of the ninth ..., 1995, pp. 2179–2186 (2005). url: http://www.uncertainty-in-engineering.net/pdf/icossar2005_pdf_017.pdf
7. Chiang, C.C., Kawa, J.: Design for Manufacturability and Yield for Nano-Scale CMOS. Springer, Netherlands (2007)
8. De Smedt, B., Gielen, G.: Watson: design space boundary exploration and model generation for analog and RF IC design. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **22**(2), 213–224 (2003). doi:[10.1109/TCAD.2002.806598](https://doi.org/10.1109/TCAD.2002.806598). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1174096>
9. Director, S., Feldmann, P., Krishna, K.: Statistical integrated circuit design. IEEE J. Solid State Circ. **28**(3), 193–202 (1993). doi:[10.1109/4.209985](https://doi.org/10.1109/4.209985). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=209985>
10. Drennan, P.G., McAndrew, C.C.: Understanding MOSFET mismatch for analog design. IEEE J. Solid State Circ. **38**(3), 450–456 (2003)
11. Floudas, C.A., Pardalos, P.M.: Encyclopedia of Optimization, vol. 1. Springer, Berlin (2008)
12. Gielen, G., Rutenbar, R.A.: Computer-aided design of analog and mixed-signal integrated circuits. Proc. IEEE **88**, 1825–1852 (2000)

13. Gong, F., Basir-Kazeruni, S., He, L., Yu, H.: Stochastic behavioral modeling and analysis for analog/mixed-signal circuits. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **32**(1), 24–33 (2013). doi:[10.1109/TCAD.2012.2217961](https://doi.org/10.1109/TCAD.2012.2217961). url: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6387696, <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6387696>
14. Herbig, V.: Getting to ‘First Time Right’ in Analog/Mixed-Signal Designs. *EE Times*, Europe (2008). url: http://www.eetimes.com/document.asp?doc_id=1271601
15. Hershenson, M.D.M., Boyd, S.P., Lee, T.H.: Optimal design of a CMOS Op-Amp via geometric programming. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **20**(1), 1–21 (2001)
16. Houck, C.R., Joines, J., Kay, M.G.: A genetic algorithm for function optimization: a matlab implementation. *NCSU-IE TR* **95**(09) (1995)
17. Houck, C.R., Joines, J.A., Kay, M.G.: A Genetic Algorithm for Function Optimization: A Matlab Implementation. Tech. rep., North Carolina State University (1996)
18. Huss, S.: Analog circuit synthesis: a search for the Holy Grail? In: 2006 IEEE International Symposium on Circuits and Systems, IEEE, pp. 1463–1466. doi:[10.1109/ISCAS.2006.1692872](https://doi.org/10.1109/ISCAS.2006.1692872). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1692872>
19. Jafari, A., Sadri, S., Zekri, M.: Design optimization of analog integrated circuits by using artificial neural networks. In: 2010 International Conference of Soft Computing and Pattern Recognition, IEEE, pp. 385–388 (2010). doi:[10.1109/SOCPAR.2010.5686736](https://doi.org/10.1109/SOCPAR.2010.5686736), URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5686736>
20. Lin, Y., Chen, D., Geiger, R.: Yield enhancement with optimal area allocation for ratio-critical analog circuits. *IEEE Trans. Circ. Syst. I: Regul. Pap.* **53**(3), 534–553 (2006). doi:[10.1109/TCSI.2005.858761](https://doi.org/10.1109/TCSI.2005.858761). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1610852>
21. Liu, B., Wang, Y., Yu, Z., Liu, L., Li, M., Wang, Z., Lu, J., Fernández, F.V.: Analog circuit optimization system based on hybrid evolutionary algorithms. *Integr. VLSI J.* **42**(2), 137–148 (2009). doi:[10.1016/j.vlsi.2008.04.003](https://doi.org/10.1016/j.vlsi.2008.04.003). url: <http://linkinghub.elsevier.com/retrieve/pii/S0167926008000126>
22. Liu, B., Fernandez, F.V., Gielen, G.: Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **30**(6), 793–805 (2011). doi:[10.1109/TCAD.2011.2106850](https://doi.org/10.1109/TCAD.2011.2106850). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5768139>
23. Mandal, P., Visvanathan, V.: CMOS Op-Amp sizing using a geometric programming formulation. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **20**(1), 22–38 (2001)
24. Martins, R.M.F., Lourenço, N.C.C., Horta, N.C.G.: Introduction. In: *Generating Analog IC Layouts with LAYGEN II*, SpringerBriefs in Applied Sciences and Technology. Springer, Berlin, pp. 1–7 (2013). doi:[10.1007/978-3-642-33146-6_1](https://doi.org/10.1007/978-3-642-33146-6_1). url: http://dx.doi.org/10.1007/978-3-642-33146-6_1
25. Mueller-Gritschneider, D., Graeb, H.: Computation of yield-optimized Pareto fronts for analog integrated circuit specifications. In: 2010 Design, Automation & Test in Europe Conference & Exhibition, IEEE, pp. 1088–1093 (2010). doi:[10.1109/DATE.2010.5456971](https://doi.org/10.1109/DATE.2010.5456971). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5456971>
26. Nye, W., Riley, D.C., Sangiovanni-Vincentelli, A., Tits, A.L.: DELIGHT.SPICE: an optimization-based system for the design of integrated circuits. *IEEE Tran. Comput. Aided Des.* **7**(4), 501–519 (1988)
27. Orshansky, M., Nassif, S.R., Boning, D.: *Design for Manufacturability and Statistical Design*. Springer, Berlin (2008). url: <http://link.springer.com/book/10.1007/978-0-387-69011-7/page/1>
28. Pelgrom MJM, Duinmaijer ACJ, Welbers APG (1989) Matching properties of MOS transistors. *IEEE J. Solid State Circ.* **24**(5), 1433–1439
29. Phelps, R., Krasnicki, M., Rutenbar, R.A., Carley, L.R., Hellums, J.R.: Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **19**(6), 703–717 (2000)

30. Razavi, B.: Design of Analog CMOS Integrated Circuits, vol 6. McGraw-Hill, New York City. doi:[10.1111/j.1151-2916.1994.tb07040.x](https://doi.org/10.1111/j.1151-2916.1994.tb07040.x)
31. Sengupta, M., Saxena, S., Daldoss, L., Kramer, G., Minehane, S.: Application-specific worst case corners using response surfaces and statistical models. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **24**(9), 1372–1380 (2005). doi:[10.1109/TCAD.2005.852037](https://doi.org/10.1109/TCAD.2005.852037). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1501902>
32. Singhee, A., Rutenbar, R.: Why Quasi-Monte Carlo is better than Monte Carlo or latin hypercube sampling for statistical circuit analysis. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **29**(11), 1763–1776 (2010) doi:[10.1109/TCAD.2010.2062750](https://doi.org/10.1109/TCAD.2010.2062750). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5605333>
33. Stefanovic, D., Kayal, M.: Structured Analog CMOS Design, 1st edn., Springer, Berlin (2008)
34. Tsividis, Y.: Operation and Modeling of the MOS Transistor, 2nd edn. Oxford University Press, Oxford (1999)
35. Xu, Y., Hsiung, K.L., Li, X., Pileggi, L.T., Boyd, S.P.: Regular analog/RF integrated circuits design using optimization with recourse including ellipsoidal uncertainty. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **28**(5), 623–637 (2009)
36. Yu, G., Li, P.: Hierarchical analog/mixed-signal circuit optimization under process variations and tuning. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **30**(2), 313–317 (2011). doi:[10.1109/TCAD.2010.2071250](https://doi.org/10.1109/TCAD.2010.2071250). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5689353>

Computational Intelligence in Analog and Mixed-Signal

(AMS) and Radio-Frequency (RF) Circuit Design

Fakhfakh, M.; Tlelo-Cuautle, E.; Siarry, P. (Eds.)

2015, XXII, 491 p. 267 illus., 174 illus. in color.,

Hardcover

ISBN: 978-3-319-19871-2