HOLMES: Capturing the Yield–Optimized Design Space Boundaries of Analog and RF Integrated Circuits

Bart De Smedt and Georges Gielen
ESAT–MICAS, Kasteelpark Arenberg 10, B–3001 Leuven, Belgium
(Bart.DeSmedt, Georges.Gielen)@esat.kuleuven.ac.be

Abstract

A novel methodology is presented to structured yield–aware synthesis. The trade–off between yield and the unspecified performances is explored along the design space boundaries, while respecting specifications on the other performances. Through the unique combination of multi–objective evolutionary optimization techniques, multi–variate regression modeling and sensitivity–based yield estimation, the designer is given access to this trade–off, all within transistor–level accuracy. Even more, a large reduction in required computer resources is obtained compared to alternative approaches.

1 Introduction

When observing the design methodology to size analog and RF integrated circuits, a transition has emerged from a 'pencil and paper' approach to a simulation–based methodology. For certain types of circuits even an optimization–based synthesis approach has become feasible. The evaluation of yield, incorporated in an automated synthesis flow, is still at an early stage of research, mainly due to computer resource limitations. This work provides in a method to explore the trade–off between yield and unspecified performances (typically power consumption), while respecting performance specifications. The accuracy of the resulting trade–off curve targeted in this work approaches transistor–level results very closely. The paper is organized as follows. In section 2, an overview of the problem and the proposed methodology is presented. Section 3 gives a brief summary of multi–objective optimization techniques. Multi–variate regression techniques are outlined in section 4. Yield calculation and trade–off analysis is discussed in section 5. In section 6 this method is applied to an operational transconductance amplifier. Finally conclusions are drawn in section 7.

2 Overview

2.1 Problem formulation

In general, a circuit design problem can formulated in the following manner[1]:

$$\begin{aligned} &\text{optimize} && \left\{\underline{\phi}_d\left(\underline{x},\underline{\xi}\right), Y\left(\underline{x},\underline{\xi}\right)\right\} \\ &\text{subject to} && \underline{\eta}\left(\underline{\zeta},\underline{\phi}_f\right) = \left[\eta_1,\eta_2,\ldots,\eta_l\right] \leq 0 \\ &\text{where} && \underline{x} = f\left(\underline{\chi},\underline{\xi}\right) \end{aligned} \tag{1}$$

The designer's task is to find the vector of nominal design variables $\underline{\chi}$, for which all functional performances $\underline{\phi}_f$ meet their specifications, all constraints $\underline{\eta}(\underline{\zeta},\underline{\phi}_f)$ are satisfied, and the combination of dissipative performances $\underline{\phi}_d$ and yield Y is optimized in some sence. $\underline{\zeta}$ represents the vector of constraint variables (e.g. drain–source voltage of a MOS transistor) and $\underline{\eta}$ is a vector of functions expressing all constraint equations and performance specifications.

Notice that the dissipative performance vector is higher–dimensional (e.g. power consumption and silicon area allocation). Moreover, an optimal value for one of the components of $\underline{\phi}_d$ does not guarantee optimal values for the other components at all. In other words, a trade–off exists between the different dissipative objectives. From a design automation point of view, however, it would be preferable to generate not a single solution to (1), but to construct the whole trade–off curve between the competing dissipative performances and yield.

Statistical process variations[2] $\underline{\xi}$ cause physical quantities of a circuit to differ from the nominal design point $\underline{\chi}$. Due to this mismatch, the circuit's functional performance deviates from the nominal performance. This might cause one or another specification to be violated; hence the circuit becomes infeasible. For this reason, aspects of yield estimation and yield optimization need to be taken into account from the very beginning.

2.2 Existing solutions

Basically two separate methodologies exist for yield–aware design automation. These differentiate by the degree of integration of the yield estimation into the design automation phase.

On the one hand, yield–aware design automation is split up in two distinctive steps: first a nominal design, followed by a local optimization to improve yield. For each of these two steps, several methods are available. [2] summarizes different research tracks targeting the nominal de-

[1] Unless otherwise stated, all optimization problems in this paper are considered to be minimization problems; translation into maximization problems is trivial.

[2] In most process fluctuation models (e.g. [7]), the variance of certain physical quantities is related to the nominal design point. Hence $\sigma_{\underline{\xi}} = f\left(\underline{\chi}\right)$.

1530-1591/03 $17.00 © 2003 IEEE

sign of analog integrated circuits. Most of these methods are based on stochastic optimization algorithms. Once the nominal design point is determined, direct yield estimation methods are used to fine–tune this nominal design point locally. These yield–estimation methods calculate the volume (or surface) integral of the (cumulative) probability density function in the disturbance space of the volume (or surface) confined within the so–called *acceptability region*. This is the set of disturbance vectors $\underline{\xi}$ for which the circuit meets all specifications, given the nominal design point [6, 11, 13]. Monte–Carlo algorithms are most suited to calculate these integrals in a high–dimensional space.

As stated in [10], gradient–based yield optimization starting from a nominal optimizer–driven design often gets obstructed in a local optimum. Also it is not always capable to find sufficiently high yield estimates. To anticipate this situation, yield estimation is integrated into the circuit synthesis step. Two examples of this methodology appeared: a single–objective simulation–based approach [10] and a single–objective symbolic approach [5]. Both methodologies result in a single design point satisfying all performance specifications and constraints and having some optimal combination of dissipative performance values and yield. However, these methods don't allow for examining the trade–off between different dissipative performances and yield. In [9] a design space boundary exploration method is presented, using successive single–objective optimization sessions. Aspects of yield were not considered. Moreover, the required computer resources are much higher than for the proposed method.

2.3 Proposed methodology

This work integrates yield–estimation in the circuit synthesis phase. As a result, the trade–off between unspecified dissipative performances and yield can be examined. The proposed design flow is depicted in Fig. 1.

1. Given the topology and technology at hand, the boundaries of the functional and dissipative performance space are explored. Samples are generated which are located at the boundaries. From these samples, the relationship between design variables and performances is retrieved within transistor–level accuracy.
2. Next the relationship between performances $\underline{\phi}$ and circuit parameters $\underline{x}$ is captured in a mathematical formulation. Not only the performances, but also the imposed constraints are captured in models for purposes that become clear later on. Here, multi–variate regression techniques are used.
3. Only at this phase, specifications are introduced according to the application in which the circuit will be plugged.[3] Specifications can either be single–sided

[3]Only at this moment, the boundaries on the functional performances

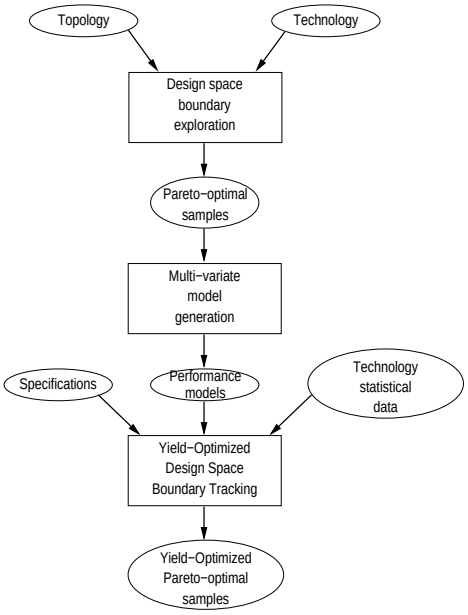

Figure 1. Proposed methodology towards yield–aware design space boundary exploration

(i.e. a lower or an upper bound) or double–sided (both upper and lower bound specified). Fig. 2 shows

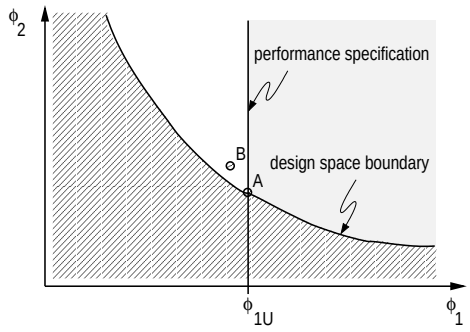

Figure 2. Illustration of yield near the design space boundaries with all performances specified

a generic case of the performance space boundaries for a functional performance ϕ_1 and a dissipative performance ϕ_2, which both need to be minimized. Also, a specification is set on ϕ_1. The white area represents the feasible design space.

4. At this phase, the trade–off between unspecified performances (typically the dissipative performances)

are known. Hence, one can judge whether or not this topology is feasible for the given set of specifications. Notice that this is often overlooked in most published design automation approaches. Typically, one imposes a set of specification on the performances, and then launches a single–objective optimization session. However, when these specifications were set too tight, the optimizer is not able to find a feasible solution.

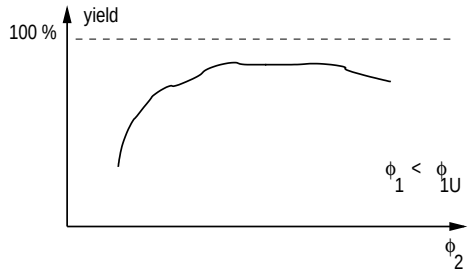

Figure 3. Trade–off between yield and dissipative performance ϕ_2

and yield is explored.[4]. E.g. by moving the optimal point A to point B, yield can be improved drastically. In this phase of the design flow, sensitivity–based yield estimation techniques are used to explore the trade–off between the unspecified performances and yield in another multi–objective optimization step. Here, the evaluations needed by the optimizer are retrieved from polynomial performance models, which result from the previous step. Fig. 3 conceptually depicts the resulting trade–off between the dissipative performance ϕ_2 and yield.

3 WATSON: exploring design space boundaries

This section depicts how the relationship between design variables and performances is explored at the boundaries of the design space. The method aims to provide results within transistor–level accuracy. Therefore a simulation–based approach is adopted in this work to evaluate the circuit performances.

3.1 Why multi–objective optimization ?

The traditional approach in analog design automation to solve the design space exploration problem formulated in (1) is the single–objective optimization technique, using a cost function in which all objectives are combined (2) [5]

$$C\left(\underline{x}\right) = \sum_{i=1}^{i=k_d} \omega_i \phi_{di}\left(\underline{x}\right) + \sum_{j=1}^{j=l} \omega_j \eta_j \left(\underline{\zeta}\left(\underline{x}\right), \underline{\phi}_f\left(\underline{x}\right)\right) \quad (2)$$

It is shown in [1] that such an approach performs only moderately for exploration purposes. In one optimization session, only a single solution is generated. Moreover, this approach is unable to capture non–convex performance space boundaries. Recent research in the domain of multi–objective optimization techniques has resulted in algorithms which are capable to track the complete design space boundary and therefore are also applicable to analog design automation to overcome the above drawbacks of the single–objective optimization methods. These techniques, which are based on evolutionary algorithms, are described in the following subsections. But first, a few new concepts are introduced.

[4] Notice that the concept of yield only becomes meaningful at this moment in the flow. Either the constraints (e.g. correct biasing of transistors) or the specifications on the functional performances might be violated when considering disturbances.

[5] Typically, this cost function consists of a penalty portion (e.g. penalty functions to steer a circuit into a feasible biasing point, or to penalize specifications below their target value) and a true objective portion (e.g. the combination of power consumption and area of an integrated circuit), all combined with proper weighting coefficients. The value of this global cost function is then minimized.

3.2 Concepts for multi–objective optimization

In an N–dimensional space, the inequality operator is redefined in (3). A decision vector $\underline{x}$ is said to be *Pareto–dominant* over another decision vector $\underline{y}$ if and only if (4) holds. Moreover a decision vector $\underline{x}$ is a *Pareto–optimal* decision vector with respect to a set A of decision vectors, if and only if $\underline{x}$ is not dominated by any of the decision vectors in the set A (5).

$$\underline{a} < \underline{b} \quad \Leftrightarrow \quad a_i < b_i \quad \forall i \in \{1, 2, \ldots, N\} \quad (3)$$

$$\underline{x} \text{ dominates } \underline{y} \Leftrightarrow \phi_i\left(\underline{x}\right) < \phi_i\left(\underline{y}\right) \; \forall i \in \{1, \ldots, k\} \quad (4)$$

$$\underline{x} \text{ is Pareto–optimal w.r.t. } A \Leftrightarrow \nexists \underline{a} \in A : \underline{\phi}\left(\underline{a}\right) < \underline{\phi}\left(\underline{x}\right) \quad (5)$$

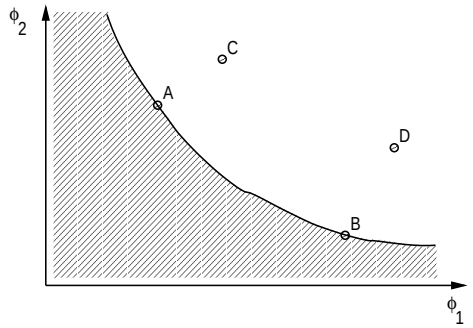

Figure 4. Pareto–optimal front

The set Σ of Pareto–optimal decision vectors for the given set A is called the *Pareto–optimal front*. An example is depicted in Fig. 4 for a two–dimensional objective space. Points A and B are elements of the Pareto–front. Their decision vectors are not dominated by any of the points in the design space. On the other hand, C is dominated by A, and D is dominated by B. Hence, neither C nor D belong to the Pareto–optimal front.

3.3 Algorithmic details

The theory of evolutionary algorithms is well established for quite some time now (e.g. [8,12]). In each iteration (*generation*), a set (*population*) of data structures (*individuals*) is created, each of which is a candidate solution to the optimization problem. The quality (*fitness*) of each individual with respect to the optimization targets is assessed first. Based on their fitness value, the individuals go through a

process of *selection*, *recombination* and finally *mutation* to constitute a new population. The overall optimization algorithm tries to tailor this population to the optimization targets. The quality of results is mostly determined by the fitness–assignment algorithm. For multi–objective optimization, the requirements on the fitness assignment are twofold. First of all, Pareto–dominant individuals should be promoted with respect to non–dominant individuals. Secondly, individuals located in a densely populated area of the design space should be demoted with respect to individuals located in sparsely populated areas in order to get uniformly distributed samples. Additionally, several flavors of stochastic operators exist which influence the convergence speed of the optimization algorithm. Both a qualitative and a quantitative comparison of these methods can be found in [4, 1, 14]. An illustration of the capabilities of this technique is shown in Fig. 5. The functional performance space (unity–gain frequency, slew rate and input–referred noise density) of two operational transconductance amplifiers (OTA), the Miller–compensated OTA and the high-speed OTA, are compared. For a certain set of specifications on the functional performances, one topology clearly is preferable over the other.

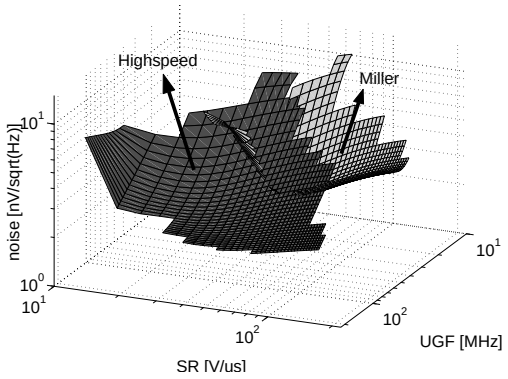

Figure 5. Topology selection

4 Multivariate regression

To capture the relationships between the design variables and the evaluated performances, a multi–variate regression technique is applied, using polynomials as basis functions:

$$\tilde{\phi}_r\left(\underline{x}\right) = \sum_{j=1}^{j=K} \alpha_{rj} \prod_{i=1}^{i=n} x_i^{\beta_{rij}} \quad \forall r \in \{1,\ldots,k\} \tag{6}$$

where

$$\alpha_{rj} \in \mathbb{R} \quad , \quad \beta_{rij} \in \{-2,-1,0,1,2\} \tag{7}$$

From the preceding optimization session, a Pareto–front set Σ containing N samples is available. In the multi–variate regression technique K degrees of freedom are introduced, where $K < N$. Substitution of the Pareto–front samples $\left\{\left(\underline{x}, \underline{\phi}\left(\underline{x}\right)\right)\right\}$ in (6) allows for solving the unknown coefficients. The complexity of this polynomial regression model is further reduced by pruning only the dominant contributions.

5 Yield–aware exploration

The final step in the proposed design flow is the trade–off analysis between unspecified performances and yield. To this end, a yield estimation method is required.

5.1 Yield calculation

Process fluctuations typically exhibit small perturbations from their nominal value. This behavior is characterized using normally–distributed probability density functions. Fig. 6 depicts the probability density function $f\left(a\right)$ of a normally distributed variable a, disturbed by process fluctuations. The combination of the mean value $\bar{a}$ and the standard distribution σ_a of the variable a fully characterize its stochastic nature.

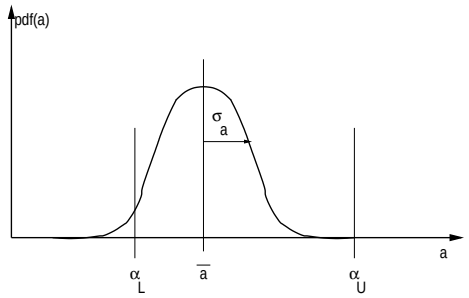

Figure 6. Normally distributed random variable with specified upper and lower bounds

After introducing specifications (upper and lower bounds), yield is calculated as the integral of the probability density function over the acceptability region (8). From a computational point of view, this definition requires a lot of computer resources to obtain accurate results in a higher dimensional design space. To overcome this problem, a set of capability indices is introduced in [3]: (9) and (10).

$$yield = \int_{a=\alpha_L}^{a=\alpha_U} f\left(a\right)\, da \tag{8}$$

$$C_p = \frac{\alpha_U - \alpha_L}{6\,\sigma_a} \tag{9}$$

$$C_{pk} = \frac{min\left(\left(\alpha_U - \bar{a}\right), \left(\bar{a} - \alpha_L\right)\right)}{3\,\sigma_a} \tag{10}$$

The C_p index is related to performance variability, whereas the C_{pk} index deals with both performance centering and performance variability. Table 1 shows the number of failures for different values of C_{pk}. When considering multiple performances simultaneously, global capability indices are defined by (11) and (12).

$$C_{p\_global} = \min_r \left(C_{p_r}\right) \quad \forall r \in \{1,\ldots,k\} \tag{11}$$

$$C_{pk\_global} = \min_r \left(C_{pk_r}\right) \quad \forall r \in \{1,\ldots,k\} \tag{12}$$

Table 1. number of failures (on a total of 1) for different C_{pk} values

C_{pk}	failures	C_{pk}	failures
0.00	0.50e0	1.00	1.35e-3
0.25	2.27e-1	1.25	8.84e-5
0.50	6.68e-2	1.50	3.40e-6
0.75	1.22e-2	1.75	7.61e-8

It is important to notice that the dimension of the design variable space $\underline{\chi}$ fundamentally differs from the dimension of the disturbance space $\underline{\xi}$.[6] Both spaces are transformed into the circuit parameter space. From that moment on, the different components of a vector $\underline{x}$ in the circuit parameter space are no longer statistically independent. Hence, the standard deviation calculation of the performances is a two–step process:

$$d\phi_r = \sum_{i=1}^{i=M} \frac{\partial \phi_r}{\partial x_i}.dx_i$$
$$= \sum_{i=1}^{i=M} \sum_{j=1}^{j=m} \frac{\partial \phi_r}{\partial x_i}.\frac{\partial x_i}{\partial \xi_j}.d\xi_j \qquad (13)$$

As the different components of the $\underline{\xi}$–vector are selected to be statistically uncorrelated, the standard deviation on the performances is calculated as:

$$\sigma_{\phi_r} = \left[\sum_{i=1}^{i=M} \sum_{j=1}^{j=m} \left(\frac{\partial \phi_r}{\partial x_i}.\frac{\partial x_i}{\partial \xi_j} \right)^2 .\sigma_{\xi_j}^2 \right]^{\frac{1}{2}} \qquad (14)$$

5.2 Yield–aware design space boundary exploration

To explore the trade–off between yield and the unspecified performances, a new multi–objective optimization session is started with the following properties.

- The optimizer takes the designable variables as decision vector (similar to the first optimization process where the design space boundaries were explored).
- The models described in section 4 are used to evaluate the performances of each individual in the population. Yield is estimated using the techniques described in the previous subsection.
- The original constraints (from the first design exploration session) are preserved as constraints.[7] Extra constraints are introduced in terms of the specified performances.
- The objectives to be optimized are the open performances as well as the capability indices C_p and C_{pk}. Note that also C_p is optimized to ensure that performance variability is reduced.

The Pareto–optimal samples, resulting from this optimization session, then represent the trade–off between the open specifications and yield.

6 Illustration: the Miller OTA

The method is now illustrated for a Miller CMOS operational transconductance amplifier (OTA) in a $0.7\ \mu m$ technology. Its topology is shown in Fig. 7. In three successive

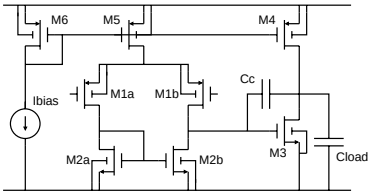

Figure 7. Miller operational transconductance amplifier

steps, the trade–off between yield and open performances is investigated.

Design space boundary exploration First the performance space boundaries are explored. The 6 performances to be explored are: power consumption [W], silicon area [m^2], gain [dB], unity–gain frequency (UGF) [Hz], slew rate [$\frac{V}{\mu s}$] and input–referred noise density [$\frac{nV}{\sqrt{Hz}}$] (evaluated at 1 kHz). The phase margin [°] is taken as constraint, together with 16 constraints on the gate–source overdrive voltage ($V_{gs} - V_T$) and drain–source voltage of all MOS transistors (to ensure proper biasing). The 14designable nominal variables for this topology are: all transistor widths and lengths, the biasing current and the Miller capacitance. Both the supply voltage and the load capacitance are considered to be fixed parameters (V_{dd} = 5.0 V and C_{load} = 10.0 pF. Furthermore, extra disturbances (mismatch on the threshold voltages, and mismatch on the current factor β, both for PMOS and NMOS devices) are taken into account at this stage, to ensure accurate model extraction in the circuit parameter space. Fig. 5 depicts the boundaries in the (UGF, SR, noise) space.

Performance model extraction The obtained Pareto front contains 828 samples. The characteristic behavior of the front is captured in a multi–variate regression model. A test population containing 200 individuals, located close to the Pareto front, is then generated to verify the model's accuracy. After model reduction, the global error measure,

[6] The first space includes all designable variables (e.g. nominal transistor dimensions, biasing currents and voltages, ...), whereas the latter space contains stochastically independent disturbances (e.g. geometrical mismatch, threshold voltage mismatch, ...) [7].

[7] The intention is twofold. First of all, these constraints are limiting the acceptability region and are therefore necessary to calculate yield. Secondly, including these constraints keeps the optimizer from screening infeasible design space areas. During the model extraction, these areas were not considered. Therefore one might expect the model accuracy to be degraded in those infeasible areas. However, by taking into account these original constraints, the optimizer is kept out of that area.

which is the overall relative standard deviation between the model evaluation and the simulated values, equals 0.021.

Yield–aware design space boundary tracking At this stage, specifications on the functional performances are introduced. In this example, the following set of specifications is introduced:

$$\text{UGF} > 75MHz, \text{SR} > 25V/\mu s, \text{gain} > 60dB$$
$$\text{input–referred noise density} < 5.0nV/\sqrt{Hz}$$
$$\text{power, area} \quad \text{to be minimized}$$

A new multi–objective optimization session is started. As initial individuals in the evolutionary population, those Pareto–front samples are selected from the WATSON session, which satisfy the introduced set of specifications. Objectives in this optimization session are the open performances power and area as well as the capability indices C_p and C_{pk}.

Table 2. Illustration summary

	performance exploration	yield exploration
generations	90	75
evaluations	9071	7525
Pareto samples	828	419
CPU time	4h 02min	3h 11min

A projection of these samples is shown in Fig. 8. The design space boundary is depicted with the straight line. As expected, by minimizing the power consumption for the given set of specifications, the design variables move closer to the design space boundaries (limited by constraints and functional specifications). Hence the yield degrades for a lower power consumption.

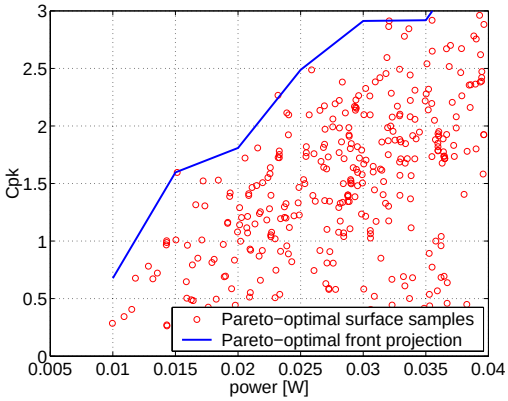

Figure 8. trade–off between yield and power consumption

As a final verification, a transistor–level simulation is carried out for some of the HOLMES Pareto–front samples. The relative difference between transistor–level simulations and model evaluations is limited within two percent. This shows that yield–aware optimization based on model evaluation has become a successful methodology.

7 Conclusions

In this paper a new approach was presented to integrate yield estimation into the nominal design flow. Using multi–objective simulation–based optimization techniques, accurate performance models are generated. In a next optimization session, the trade–off between sensitivity–based yield calculation and unspecified performances is explored within transistor–level accuracy. The Miller OTA illustrates this novel methodology.

References

[1] B. De Smedt and G. Gielen. Watson: Design Space Boundary Exploration and Model Generation for Analog and RF IC Design. *IEEE Transactions on Computer–Aided Design of Integrated Circuits and Systems*, 22(2), February 2003.

[2] L. Carley, G. Gielen, R. Rutenbar, and W. Sansen. Synthesis Tools for Mixed–Signal ICs : Progress on Frontend and Backend Strategies. In *DAC*, June 1996.

[3] J. Chen and M. Styblinski. IC variability minimization using a new Cp and Cpk based variability/performance measure. In *International Symposium on Circuits and Systems (ISCAS)*, June 1994.

[4] C. Coello. *An Empirical Study of Evolutionary Techniques for Multi–objective Optimization in Engineering Design*. PhD thesis, Tulane University, 1996.

[5] G. Debyser and G. Gielen. Efficient Analog Circuit Synthesis with simultaneous Yield and Robustness Optimization. In *International Conference on Computer–Aided Design (ICCAD)*, November 1998.

[6] S. Director, P. Feldmann, and K. Krishna. Statistical Integrated Circuit Design. *JSSC*, 28(3):193–202, March 1993.

[7] P. Drennan and C. McAndrew. Understanding MOSFET mismatch for analog design. In *Custom Integrated Circuits Conference (CICC)*, pages 23.4.1–23.4.4, May 2002.

[8] D. Fogel. An introduction to simulated evolutionary optimization. *T. on Neural Networks*, 5(1):3–14, January 1994.

[9] R. Harjani and J. Shao. Feasibility and Performance Region Modeling of Analog and Digital Circuits. *Analog Integrated Circuits and Signal Processing*, 10:pp. 23–43, 1996.

[10] T. Mukherjee, L. Carley, and R. Rutenbar. Efficient handling of operating range and manufacturing line variations in analog cell synthesis. 19(8):825–839, 2000.

[11] F. Schenkel, M. Pronath, S. Zizala, R. Schwencker, H. Graeb, and K. Antreich. Mismatch Analysis and Direct Yield Optimization by Spec–Wise Linearization and Feasibility–Guided Search. In *DAC*, June 2001.

[12] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994.

[13] J. Zhang and M. Styblinski. *Yield and variability optimization of integrated circuits*. Kluwer Ac. Publ., 1995.

[14] E. Zitzler. *Evolutionary Algorithms for Multi–objective Optimization : Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology Zurich, 1999.