ITRS 2011 Analog EDA Challenges and Approaches

(Invited paper)

Helmut Graeb
Institute for Electronic Design Automation
Department of Electrical Engineering and Information Technology
Technische Universität München, Munich, Germany

***Abstract*—In its recent 2011 version, The International Technology Roadmap for Semiconductors [1] updated a section on analog design technology challenges. In the paper at hand, these challenges and exemplary solution approaches will be sketched. In detail, structure and symmetry analysis, analog placement, design for aging, discrete sizing, sizing with in-loop layout, and performance space exploration will be touched.**

Keywords-analog design, placement, layout, sizing, reliability, aging, yield, Pareto, optimization, synthesis

I. INTRODUCTION

Sophisticated algorithms for analog design have been published since the 1970ies. It took about 30 years until analog synthesis tools have become part of commercially available EDA tools, but hardly any analog EDA tool on the market beyond circuit simulation is established in practice. This is a serious problem, as analog components are on virtually every chip: even a chip considered purely digital will have, e.g., a clock generation component, mostly a mixed-signal phase-locked loop circuit. Considering a study of the IBS Group, cited in 2005 in EDA Weekly [2], on average 20% of the chip area are analog, but 40% of the design effort and 50% of the costly design re-spins are due to analog components. It can be assumed that these numbers are valid on a long term. They correspond to the persisting expensive and error-prone manual process of analog circuit design.

Obviously, there is a deadlock situation, where on the one hand more EDA in analog design is required to increase efficiency and quality of the design, and where on the other hand it has not possible to establish analog EDA beyond simulation. In [1], few overall recommendations to escape this deadlock and several specific challenges of analog EDA are given, which will be discussed in the rest of the paper.

II. ANALOG IN SYSTEM DESIGN

Analog circuits are components of complex systems, realized on one chip. Fig. 1 gives an impression of the various components that appear on such a system on a chip (SoC). Depending on the embedding of the chip, i.e., on its application, different parts will play more dominant roles than others. Analog components appear at various place, e.g., as cells of an SRAM, within voltage regulation and power-on reset, as signal converter between analog (environment) and digital (signal processing). System synthesis traverses different design views and abstraction levels (Fig. 2). Synthesis is usually attributed to the transit from the behavior

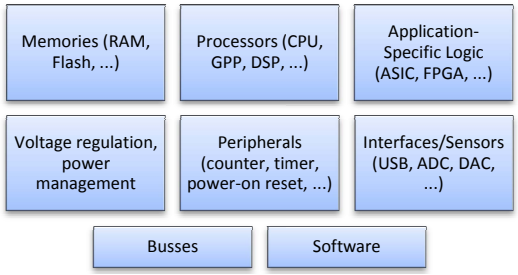

Figure 1. System on a chip: components

view of the system specification over the structure view, e.g., by netlists, to the geometrical view of the chip layout. The transit from the overall system level to more detailed implementations of system blocks like processors in logic (gates) and circuits (transistors) is usually denoted as refinement [3], [4].

For the hardware parts of system design with dedicated algorithmic functions, well-rehearsed design processes exist in form of sequential steps of high-level synthesis, logic synthesis, layout synthesis. System design starts from an algorithmic description, e.g. in SystemC, creates through high-level synthesis steps of scheduling, binding and allocation an HDL model of controller, data path and interface, which becomes a gate netlist after logic synthesis and mask data after layout synthesis.

For an SoC with software and hardware components in submicron technologies, the design situation is more complex. Design prototyping and design partitioning require a proper preparation of the design process and profound modeling of power, timing and area of underlying hard and soft macro libraries. At system level, the industrial design process is to a large extent still manual.

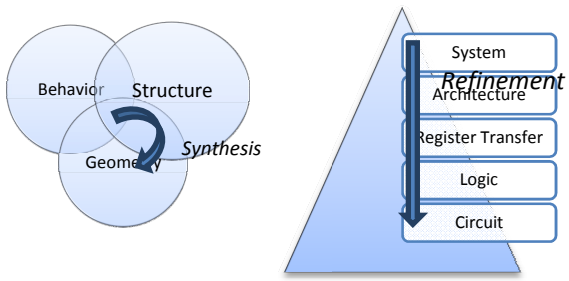

Figure 2. System synthesis: design views (left) and abstraction levels (right)

978-3-9810801-8-6/DATE12/©2012 EDAA

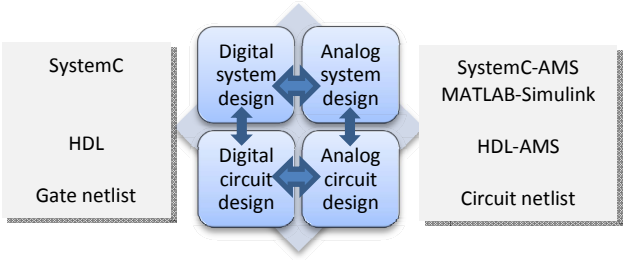

Figure 3. Analog and digital system synthesis and typical modeling methods on different abstraction level

On the other hand, design decisions on system level affect the analog components and vice versa. Analog design exploration must therefore start on system level *together with* the rest of the system, *not after* the digital architecture has been generated. System design exploration is therefore a simultaneous process for digital and analog components (Fig. 3). This is also reflected by analog mixed-signal extensions of the main modeling languages.

This means that analog design is at the side of digital design from the top of the refinement pyramid in Fig. 2 to the bottom and does *not* just start when digital design reaches the bottom at the circuit level! This strong interrelation is an important reason according to [1] why analog design is strongly manual. It makes the decision where to put analog/mixed-signal (AMS) elements like converters or mixer into the signal chain very difficult, as well as the decision how to trade-off which performance objectives [1]. Instead, methods and tools for *interactive and semiautomatic* AMS design should be the goal. Against this background, modeling and simulation of complex analog/mixed-signal/RF (AMSRF) blocks is put in the first place of EDA tasks to be done.

III. Simulation of Complex AMSRF blocks

We will now have a look at three examples of system blocks with analog component and discuss cost and feasibility of simulation.

A. Operational Amplifier

Operational amplifiers like the folded-cascode structure in Fig. 4 are purely analog and can be simulated easily. Transistor netlists, transistor models including statistics and set-up for automatic simulation are available for everyone. Transient simulation takes a few seconds.

B. Phase-Locked Loop

Phase-locked loops as shown in Fig. 5 are mixed-signal circuits with charge pump, loop filter and oscillator as analog parts, and phase-frequency detector and divider as digital parts.

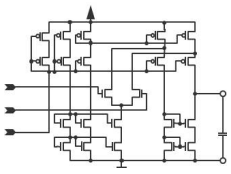

Figure 4. Folded-cascode operational amplifier

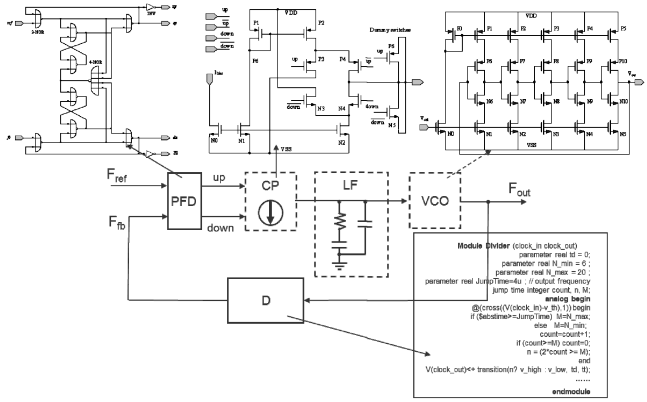

Figure 5. Circuit model of phase-locked loop: phase-frequeny detector (PFD), charge pump (CP) and voltage-controlled oscillator (VCO) as transistor netlist, divider (D) as HDL code

These circuits cannot be simulated as easily as Opamps. If they are modeled in the same way as Opamps, i.e., taking transistor netlists as in Fig. 5, and using the available transistor models, simulation of features like damping constant and locking time takes more than 2 days, simulation of the frequency even a week!

In order to be able to simulate a PLL, more abstract models, e.g. using a hardware description language (HDL) are required. Fig. 6 shows a block-based modeling of this PLL. Simulation of the frequency now takes in the range of minutes.

However such models are not available automatically and must be created by hand. And the problem of getting from the HDL models down to the transistor sizing remains open as another task that has to be solved by hand.

C. Analog Receiver Front End

The upper left part of Fig. 7 shows a schematic of a communication system, where at the output of the multipath channel a simple arrow denotes the analog receiver front end, whose block level schematic is given in the upper right part of the figure. For two important components of the receiver front, mixer and low noise amplifier, the transistor netlists are given. It is not possible today to simulate the whole receiver front end from the transistor level up to its system properties, not to mention of closing the link to the analysis and design of the complete communication system!

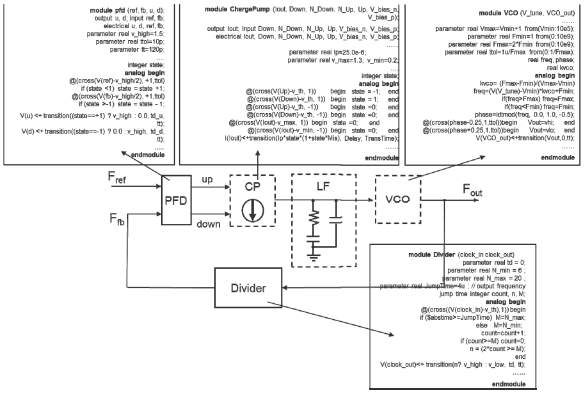

Figure 6. HDL model of phase-locked loop based on [5]

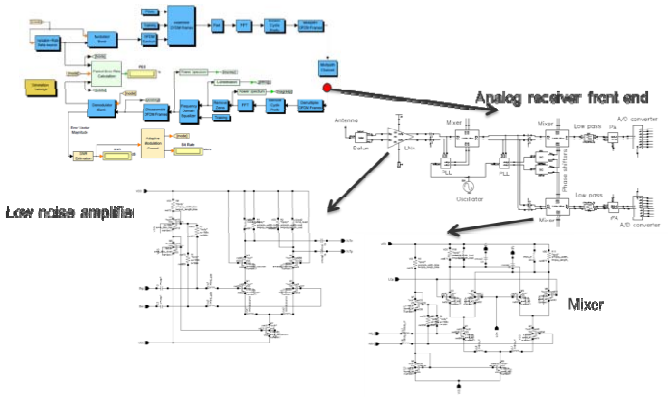

Figure 7. Communication system with analog receiver front end and its basic components mixer and low-noise amplifier

IV. ITRS 2011 REQUIREMENTS ON MODELING AND SIMULATION OF AMS CIRCUITS

The three examples in section III correspond to the three abstraction levels sketched on the right side of Fig. 3. They illustrate the requirements on modeling and simulation of AMS circuits mentioned in [1]. There are standards for modeling circuits on each of the three levels, which are determined by the respective modeling languages and corresponding simulation methods. But there is no seamless hierarchical way to move across the abstraction levels in a viable way in industrial practice, neither bottom up for analysis and verification, nor top down for synthesis and optimization (Fig. 8). Closing the gap between the abstraction levels for a seamless hierarchical design is the "number one" requirement for further development of EDA methods for AMS design [1]. Practicable solution approaches must take into account several paradigms [1]:

A. *Standardized Blocks and Hierarchical Parametric Block Models*

The MOS transistor is established as the standard basic device of integrated circuits and exemplary for moving the design across levels of abstraction. By modeling a circuit as a netlist of transistors, its behavior can be defined by certain performances that are simulated with SPICE-like numerical methods. The next lower level of abstraction is the device and manufacturing level, which is captured through a set of transistors. This set contains NMOS, PMOS, small or large geometries etc. The structure of the transistor model is standardized through, e.g., BSIM [6] or EKV [7], and contains a variety of parameters, some of which are determined beforehand for the underlying manufacturing process, and some of which are determined during design.

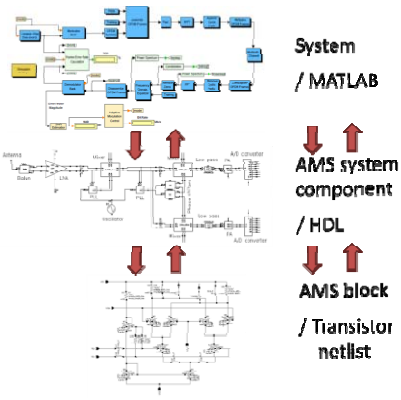

Figure 8. Hierarchical AMS design: simulation (bottom up) and synthesis (top down)

The power of the transistor device model lies in its immediate reusability in any type of circuit. A comparable automatism to higher levels of abstraction than the circuit level still does not exist. In order to be able to make full use of the meanwhile powerful modeling methods in VHDL/Verilog-AMS and SystemC-AMS within a hierarchical design process, standardized circuit blocks have to be defined.

The open question is: What can become the correspondent of a transistor on a higher level of abstraction, what type of more abstract component has a comparable potential for reusability in a general design flow?

Independently of that big question, we need a systematic development of currently prevalent components like receiver front ends, A/D and D/A converters, Opamps towards standardized components with exchangeable models on different levels of abstractions. This requires a tedious job of standardizing types of blocks, numbers and types of pins, simulation benches, parameters, automatic simulation etc.

The ideal goal should be to be able, e.g., to plug the analog receiver front end into a schematic of a communication system, and further detail its blocks with different transistor netlist implementations for, e.g., mixers, amplifiers. The availability should be open source or at least through an EDA vendor, rather than specific in-house solutions.

B. *Simulation During Design, Not After Design*

The state of the art in analog design is to use simulation after having completed the design in order to verify it. The design is based on manually controlled explorations of structural alternatives and parametric trials or sweeps. Circuit behavior is modeled through a composition of the basic transfer behavior of basic building blocks like differential stages. Block identification and behavior composition is mostly manual in practice.

In order to boost the design productivity, a change towards applying simulation within a semi-automatic synthesis and optimization process is needed. This includes several aspects:

1) Put simulation at the beginning of the design process, not the end: This means that a design task is first prepared in such a way that all or most or the most important aspects can be simulated. Only at first sight this seems to be a loss of time. Actually, the design problem gains access to powerful optimization approaches, without loosing its physical roots. This saves a lot of design time as more complex tasks can be handled less error-prone.

2) Simulation Shell Standards for Blocks: An international standard should be developed that defines

a) a set of relevant analog and mixed-signal blocks, starting from Opamps to bigger blocks,

b) complete sets of performance features for each block, like, e.g., PSRR, CMRR, noise figures, etc,

c) definitions of simulation benches for each performance feature,

d) definitions of extraction routines to compute each performance feature from the output waveform of its corresponding simulation.

Standardization should include circuit design knowledge beyond the pure modeling languages. Such a standard must consider the hierarchy within AMS systems. It will also need a discussion how to combine/consider a *gm/Id*-centric and a *W&L*-based view on sizing. Based on such a standard, AMS design automation could be enhanced from the current industrial state of the art, where simulation waveforms are often evaluated manually.

3) Automatic sensitivity calculation of all performance features: Sensitivities are a powerful tool to gain insight into the dependencies and correlations among multiple parameters and multiple objectives of a design problem and provide gradients for optimization. For the performance features standardized as described before, an automatic sensitivity calculation with regard to any simulator input parameter (transistor model or geomeetry parameter, temperature, supply voltage etc) must be available. This task refers to standardization as well as to simulator capabilities. Simulators should be able to perform sensitivity calculation for any type of simulation, AC, DC, TR, HB, etc, and simulators and standards should enforce that models attached to the simulator by the user include the required sensitivity equations.

4) Curriculum of circuit design and EDA development study programs: A feature of the described requirements is that a closer interaction between EDA for and design of analog/mixed-signal circuits is needed. On one side, tools and algorithms need to incorporate circuit design knowledge more deeply. On the other side, design needs to incorporate more formal methods. Another sign of the closer interaction is the requirement for interactive and semi-automatic tools. Programmers must know about circuit design to create the right interrupts for user input in their tools, and designers must know about mathematical algorithms to give the right input. As a consequence, university curricula should provide and recommend a combination of courses on analog/mixed-signal design and on numerical mathematics and optimization. A specialization that leaves out one side, either the design side, or the optimization side, is a big drawback.

V. ITRS 2011 Requirements on Synthesis and Optimization of AMS Circuits

The ITRS 2011 edition itemizes some specific requirements on tools for synthesis and optimization of analog/mixed-signal circuits, which are [1]:

- interactive generation and selection of circuit structures,
- interactive placement and routing,
- interactive design for yield and reliability,
- discrete optimization,
- integration of structural and layout synthesis,
- design space exploration.

Problem formulations and solution approaches referring to these tasks will be sketched in the following sections.

VI. Synthesis and Optimization

Structural synthesis tools have not been able to prove their practicability so far. Recent approaches concentrate on generation and selection of available structures along a circuit hierarchy within design space exploration [8], [9]. Taking up the ITRS recommendation on stronger interactivity of the design tools, it is worth to focus on methods for a thorough *analysis* of inherent hierarchical structures of circuit classes rather than synthesis. In [10], such a method is presented to compute the internal hierarchy and the qualitative signal flow in Opamps (e.g., fully differential Opamp in Fig. 9).

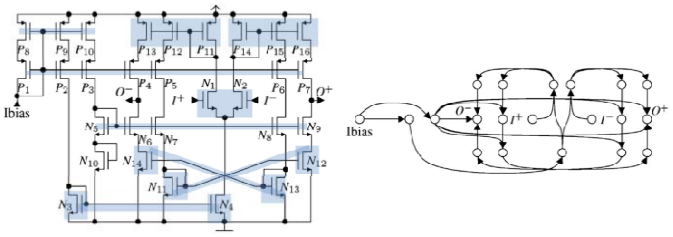

Figure 9. Structural hierarchy and signal flow ([10] © IEEE 2011)

Please note that the result of structural and behavioral analysis sketched in Fig. 9 is fully compatible with the result of a manual inspection of the same circuit netlist. Compatibility of automatic design algorithms with manual design processes is a necessity that immediately follows from the requirement on interactive design algorithms: the tool must be "fluent in the "language" of designers not only in the beginning and at the end, but throughout the whole design process. A further inevitable consequence of interactivity and compatibility then is a shift from statistical to deterministic/constructive solution methods. Statistical solution methods are adequate only if detailed knowledge on a process is not applied. On the contrary, the manual AMS design is rooted in a specific, sophisticated proceeding that has to be captured in a deterministic/constructive algorithm in order to provide the required interactivity and compatibility. An example is the application of the method [10] within a new constructive algorithm for analog placement [11].

VII. Placement and Routing

Analog layout is mostly done manually today. Even more than other analog design tasks, it requires strongly application-specific solution approaches, which makes its automation very difficult. Some recent approaches to analog layout synthesis can be found in [12], [13]. Among current analog layout approaches, the method [11] stands out for extensively reproducing the manual proceeding of analog layout design. It automatically generates both comprehensive layout constraints and a placement flow from the inherent hierarchy in circuit netlists. Fig. 10 illustrates the hierarchical placement flow and layout that result for the circuit in Fig. 9. There is a hierarchy of groups of devices that have to comply with matching from building blocks or symmetric signals (MG_B or MG_S), symmetry (SG) and proximity from building blocks or the whole netlist (PG_B or PG_N) constraints. This hierarchy can be used as the "design plan" for computing the placement in a constructive, bottom-up way fully compatible with the manual proceeding.

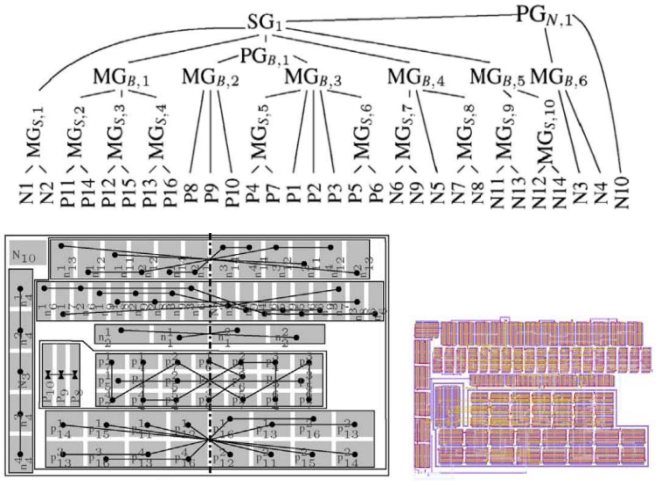

Figure 10. Hierarchical placement flow and resulting layout of the circuit in Fig. 9 ([10] © IEEE 2011)

For the leaf groups, e.g. $MG_{S,x}$ in Fig. 10, all feasible placements are enumerated. For the higher-level groups, placements are subsequently constructed by efficiently merging the lower-level placements [11]. The search space is mainly bounded by the inherent hierarchy, no statistical search is required! As a result, all possible aspect ratios of a circuit are provided. The method works fast for usual analog circuit sizes around 100 transistors.

Please note that solution algorithms for structural synthesis, sizing and layout synthesis have two features: one is the choice and development of numerical optimization algorithms, the other is the method to capture design knowledge and translate it into optimization objectives, optimization constraints and control mechanisms of the optimization process. The latter should enable semiautomatic and interactive optimization approaches.

VIII. Design for Yield and Reliability

Yield [14], [15], [16] and, more recently, reliability [17], [18] are important issues in analog/mixed-signal circuit design. There are sophisticated methods to analyze and maximize yield, available as commercial tools, e.g. WiCkeD [19]. The focus of current research on analog reliability is on modeling and simulation of aging effects on transistor level. Some research has been done on methods for reliability optimization of analog circuits [20]. From the results so far, analog reliability optimization should deal with the following features and questions:

1. Which circuits and performances are aging-critical?
2. Aging and reliability should be considered during sizing.
3. Reliability and yield optimization should be integrated.
4. Sizing with reliability constraints is faster than reliability optimization with aging simulation.

Item 3 has been targeted in [20] with the concept of lifetime yield and life-time worst-case distances. The basic idea of worst-case distances [21] was to approximate the yield Y over linear performance models in the finite statistical parameter element $\boldsymbol{s}_{w,i}$ with highest probability to violate the spec $f_{B,i}$, for all specs i:

$$\beta_{w,i}^2 = \left(\boldsymbol{s}_{w,i} - \boldsymbol{s}_0\right)^T \boldsymbol{C}^{-1} \left(\boldsymbol{s}_{w,i} - \boldsymbol{s}_0\right)$$

$$\boldsymbol{s}_{w,i} = \arg\min_{\boldsymbol{s}} \{\beta^2(\boldsymbol{s}) \big| f(\boldsymbol{s}) = f_{B,i}\}$$

$$Y_i = \int_{-\infty}^{\beta_{w,i}} \frac{1}{\sqrt{2\pi}} e^{-\zeta^2/2} d\zeta, \qquad 1 - \sum_i (1 - Y_i) \le Y \le \min_i Y_i \ ,$$

with $\boldsymbol{s}_0$ being the mean value vector and $\boldsymbol{C}$ the covariance matrix of statistical parameters $\boldsymbol{s}$. β_w is called worst-case distance. Aging effects are considered by extending all variables that are affected by aging to become time-dependent variables: $\boldsymbol{s}_0 \rightarrow \boldsymbol{s}_0(t)$; $\boldsymbol{C} \rightarrow \boldsymbol{C}(t)$; $\beta_{w,i} \rightarrow \beta_{w,i}(t)$;

$$\boldsymbol{s}_{w,i} \rightarrow \boldsymbol{s}_{w,i}(t); Y_i \rightarrow Y_i(t)$$

Fig. 11 illustrates this for two statistical parameters.

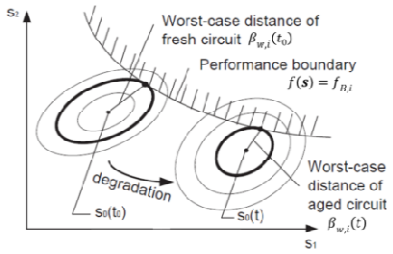

Figure 11. Aging leads to a degradation of the mean values (and covariances) of statistical parameters. This leads to a decrease in the worst-case distances and yield over lifetime.

A basic reliability optimization flow is shown in Fig. 12. The flow has been made more efficient by, first, reduced calls of aging simulation, second, applying a predictive model of lifetime worst-case distance, third, sizing rule check for aged circuits (DC aging simulation) [20]. More research on the mentioned items 1 to 4 is required.

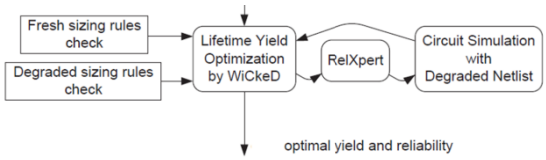

Figure 12. Basic reliability optimization with lifetime yield

IX. Discrete Optimization

Most existing sizing tools are based on continuous optimization. Most of the design parameters, however, are actually discrete. For instance, transistor geometries must snap to a given manufacturing grid. Many existing analog sizing tools handle discrete parameters by continuous optimization with subsequent rounding, which is known to be suboptimal [22]. A hybrid algorithm using Nonlinear Programming and Branch-and-Bound has been developed as a first step [23]. More research is required to include parameter tolerances [24], and to consider that only discrete parameter values can be simulated. Beyond that, discrete analog optimization must also be enabled to consider *unordered* optimization variables, i.e., parameters where no Taylor-like performance models can be established.

X. Integration of Structural & Layout Synthesis

Especially RF circuits cannot be sized without simultaneous consideration of the parasitic effects from layout. In deeper submicron technologies, layout effects more and more have to

be considered for regular types of analog/mixed-signal circuits as well. This can be achieved by integrating layout synthesis and sizing. For efficiency reasons, layout templates are frequently applied to model the layout effects within sizing [25], [26], [27]. An interesting approach has been published in [28]. Here, the fast deterministic placement approach [11] is integrated with the sizing tool [19] to consider the *true* layout. The runtime is still acceptable: it increases by a factor of roughly 8 compared to sizing with layout consideration. This integration requires a sophisticated control of layout generation. More research and development is needed to come to practicable solutions and tools.

XI. Design Space Exploration

Design space exploration is a very important part of a semi-automatic/interactive design flow. Trade-offs between different performance objectives and among different implementation variants, considering parameter tolerances, require sophisticated solution approaches based on Pareto optimization [27], [29], [30], [31], [32], [33], [34]. In [35], the first approach to systematically span a multi-dimensional Pareto front has been introduced and illustrated with analog circuits. In [36], an advanced method to compute yield-optimized Pareto fronts of analog circuits has been presented. Fig. 13 illustrates a typical situation for two performances of an Opamp. The axes are such that better circuits are in direction of the lower left corner. The three fronts given from lower left to upper right are: nominal front, new approach for yield-optimized front, state-of-the-art yield-aware front. It can be seen that neglecting the yield suggests a much too optimistic performance and that the new approach leads to a much better performance under a given yield constraint. More research to deal with the issue of simulation cost in Pareto optimization and to consider various other trade-offs, e.g., considering reliability, is required.

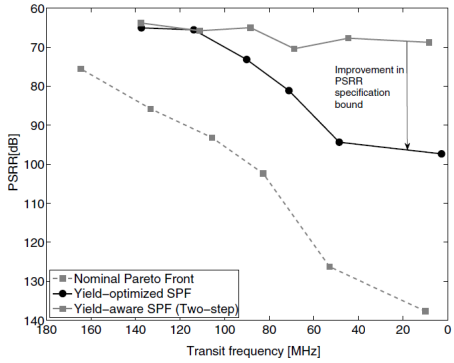

Figure 13. Pareto fronts of some opamp ([36] © EDAA 2010).

XII. Conclusion: Is There A Business (Research) Model for Analog EDA?

This paper has illustrated requirements on analog/mixed-signal design tools mentioned in the ITRS roadmap [1]. The main messages are that modeling/simulation of complex systems is urgently missing, and that algorithms and tools need to be compatible with the manual design process. An important question that comes up as so many analog EDA start-up tools disappear after a while is whether there can be a business (and also research) model for analog EDA. My answer is: yes, if it completes tools with proper consulting, as demonstrated by, e.g., Designer's Guide Consulting [5], MunEDA [19], or PDF Solutions.

References

[1] http://www.itrs.net/
[2] J. Horgan, "Analog", EDA Weekly, March 2005, http://www10.edacafe.com/nbc/articles/2/209175/Analog
[3] G. De Micheli, Synthesis and Optimizatin of digital circuits, McGraw-Hill, 1994.
[4] D. Gajski, S. Abdi, A. Gerstlauer, G., Schirner, Embedded System Design, Springer, 2009.
[5] K. Kundert, H. Chang, http://www.designers-guide.com/
[6] http://www-device.eecs.berkeley.edu/~bsim3/
[7] http://ekv.epfl.ch/
[8] T. McConaghy, P. Palmers, M. Steyaert, G. Gielen, Variation-Aware Structural Synthesis of Analog Circuits via Hierarchical Building Blocks and Structural Homotopy, IEEE Trans. CAD, 2009.
[9] O. Mitea, M. Meissner, L. Hedrich, Topology Synthesis of Analog Circuits with Yield Optimization and Evaluation using Pareto Fronts, IEEE Int. Conf. VLSI and System-on-Chip, 2011.
[10] M. Eick, M. Strasser, Kun Lu, U. Schlichtmann, H. Graeb: Comprehensive Generation of Hierarchical Placement Rules for Analog Integrated Circuits, IEEE Trans. CAD, 2011.
[11] M. Strasser, M. Eick, H. Graeb, U. Schlichtmann: Deterministic Analog Placement by Enhanced Shape Functions, in: Analog Layout Synthesis – A Survey of Topological Approaches, Springer, 2011
[12] H. Graeb (Editor): Analog Layout Synthesis – A Survey of Topological Approaches. Springer, 2011.
[13] E. Yilmaz, G. Dündar, Analog Layout Generator for CMOS Circuits, IEEE Trans. CAD, 2009.
[14] H. Graeb, Analog Design Centering and Sizing, Springer, 2007.
[15] A. Singhee, R. Rutenbar, Statistical Blockade: Very Fast Statistical Simulation and Modeling of Rare Circuit Events and Its Application to Memory Design, IEEE Trans. CAD, 2009.
[16] B. Liu, F. Fernandez, G. Gielen, Efficient and Accurate Statistical Analog Yield Optimization and Variation-Aware Circuit Sizing Based on Computational Intelligence Techniques, IEEE Trans. CAD, 2011.
[17] E. Maricau, G. Gielen, Efficient Variability-Aware NBTI and Hot Carrier Circuit Reliability Analysis, IEEE Trans. CAD, 2010.
[18] E. Maricau, G. Gielen, Computer-Aided Analog Circuit Design for Reliability in Nanometer CMOS, IEEE JETCAS, 2011.
[19] www.muneda.com
[20] Xin Pan, H. Graeb, Lifetime Yield Optimization of Analog Circuits Considering Process Variations and Parameter Degradations, In: Advances in Analog Circuits, InTech, 2011.
[21] K. Antreich, H. Graeb, Circuit optimization driven by worst-case distances, IEEE Int. Conf. on Computer-Aided Design (ICCAD), 1991.
[22] G. Nemhauser, L. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons, 1988.
[23] M. Pehl, H. Graeb, An SQP and Branch-and-Bound Based Approach for Discrete Sizing of Analog Circuits, In: Advances in Analog Circuits, InTech, 2011.
[24] M. Pehl, H. Graeb, Variability Aware Automated Sizing of Analog Circuits Considering Discrete Design Parameters, IEEE Int. Symp. on Integrated Circuits (ISIC), 2011.
[25] R. Castro-Lopez, O. Guerra, E. Roca, F. Fernandez, An Integrated Layout-Synthesis Approach for Analog ICs, IEEE Trans. CAD, 2008.
[26] A. Unutulmaz, G. Dündar, F. Fernandez, A Template Router, Europ. Conf. Circuit Theory & Design (ECCTD), 2011
[27] A. Toro-Frias, R. Castro-Lopez, E. Roca, F. Fernandez, Layout-Aware Pareto Fronts of Electronic Circuits, ECCTD, 2011.
[28] H. Habal, H. Graeb, Constraint-based Layout-driven Sizing of Analog Circuits, IEEE Trans. CAD, 2011.
[29] B. Liu, F. Fernandez, Q. Zhang, M. Pak, S. Sipahi, G. Gielen, An Enhanced MOEA/D-DE and Its Application to Multiobjective Analog Cell Sizing, IEEE CEC, 2010.
[30] G. Yu, P. Li, Hierarchical Analog/Mixed-Signal Circuit Optimization Under Process Variations and Tuning, IEEE Trans. CAD, 2011.
[31] E. Deniz, G. Dündar, Hierarchical Performance Estimation of Analog Blocks using Pareto Fronts, PRIME, 2010.
[32] O. Mitea, M. Meissner, L. Hedrich, Topology Synthesis of Analog Circuits with Yield Optimization and Evaluation using Pareto Fronts, VLSI-SoC, 2011.
[33] G. Stehr, H. Graeb, K. Antreich: Analog Performance Space Exploration by Normal-Boundary Intersection and by Fourier-Motzkin Elimination, IEEE Trans. CAD, 2007.
[34] S. Tiwary, P.Tiwary, R. Rutenbar, Generation of Yield-Aware Pareto Surfaces for Hierarchical Circuit Design Space Exploration, ACM/IEEE DAC, 2006.
[35] D. Mueller-Gritschneder, H. Graeb, U. Schlichtmann: A Successive Approach to Compute The Bounded Pareto Front of Practical Multi-Objective Optimization Problems, SIAM Journal on Optimization (SIOPT), 2009.
[36] D. Mueller-Gritschneder, H. Graeb, Computation of Yield-optimized Pareto Fronts for Analog Integrated Circuit Specifications, Design, Automation, and Test in Europe (D.A.T.E) Conference, 2010.