

Near-Term Industrial Perspective of Analog CAD

Christopher Labrecque

Synopsys, Inc.
Mountain View, CA
940043, USA

chris.labrecque@synopsys.com

ABSTRACT

Analog and mixed-signal CAD looks like a nice success story: there's been significant research in building design automation tools since the late 80's, and commercial tools have been on the market for several years now. However, the majority of AMS (Analog/Mixed-Signal) designers still use manual design only, focused around the SPICE simulator. So why are designers not or slowly adopting these CAD tools? This paper will present a reality check on the current state of the art of AMS design tools for industrial usage.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design aids –*graphics, layout, placement and routing, simulation, verification*

General Terms

Algorithms, Design, Verification.

Keywords

Analog, mixed-signal, integrated circuits, computer-aided design

1. Introduction

Many believe analog circuit design has not changed in 30 years. Is this true? Compared to the digital design revolution, yes; analog design has not changed nearly as much as its digital counterpart. That said, there has been, and will continue to be, evolutionary advancements in analog design automation.

First, let's ask: What does an analog CAD tool need to do to gain industry adoption? The answer is the same with any technological innovation: The value of the tool must be well worth the incremental effort in setting up the design problem. In addition, automated techniques must efficiently solve real design problems, problems that are difficult to solve with manual design techniques. If the manual methodology works, why fix it?

Looking back, advancements in analog design tools have played a critical role in increasing the efficiency of analog circuit design and verification. Going forward, the demand for automated analog design technology will be driven by the challenges of shrinking

semiconductors and increasing board and chip speeds.

Let's review a few areas of analog CAD, discuss why certain techniques have been adopted, discuss why others have not, and attempt to forecast which automation techniques will be vital in meeting the near-term future challenges of analog design.

1.1 The SPICE Advance

Current Status: Starting from roughly the early 1970s, SPICE and SPICE models started to become adopted by analog designers, to their now longtime stature as indispensable design tools. Before SPICE, analog circuit analysis and design was done mostly by hand, by solving equations on paper. SPICE complemented these manual calculations, providing designers with an accurate way to verify their designs, and enabling designers to rapidly understand the effects of process and environmental conditions. SPICE has advanced considerably over the past 30 years. The basic AC, DC, transient and steady-state simulation techniques are much faster and more accurate, and the addition of new analysis techniques has enabled designers to create circuits that operate at high speeds, are tolerant to noise, and have high yield.

The Near Future: Chip and board speeds are increasing. Consequently, advanced signal integrity analysis and models are becoming part of mainstream design. High-speed phase-locked loops are becoming more popular, requiring designers to use RF-styled analysis such as Harmonic Balance to simulate phase noise and jitter effects. As semiconductor device sizes shrink, process variability analyses are becoming increasingly popular with both Integrated Device Manufactures (IDMs) and fabless design houses.

1.2 Fast SPICE and Behavioral Modeling

Current Status: Behavioral modeling has become very popular for testbench development. Designers frequently use the behavioral representations in languages such as Verilog-A, Verilog-AMS and VHDL-AMS to model complex input stimuli. Behavioral models are also used in design when fast turnaround is desirable and rough accuracy is acceptable, such as exploration of system-level architectures. They can be used for bottom-up verification as well, but here they share a usage environment with FastSPICE simulators. Though behavioral models simulate faster, they require designer setup time and do can compromise considerable accuracy; FastSPICE doesn't have big requirements in designer setup time and has better accuracy (albeit at longer simulation times). Generality, accuracy and runtime of FastSPICE simulators has improved greatly in recent years, to the point where they are now used to simulate full chips at the transistor level; as a result FastSPICE simulators have an enviable adoption record. Behavioral modeling and FastSPICE aren't necessarily mutually exclusive either: designers can choose to use a mix, essentially turning the dial on speed vs. accuracy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD'06, November 5-9, 2006, San Jose, CA

Copyright 2006 ACM 1-59593-389-1/06/0011...\$5.00

The Near Future: Designers will increasingly employ behavioral modeling for testbench development. Transistor-level FastSPICE simulation will remain popular, while behavioral modeling will gain usage where simulation speed is more critical than simulation accuracy.

1.3 Front-End Performance Optimization

Current Status: Performance optimization at the transistor level is becoming increasingly popular for designing high performance analog designs, for migrating analog and custom digital designs to new process technology nodes, and for creating high-speed, low-power digital cell libraries. Since performance optimization requires up-front setup of constraints, its adoption for lower-performance analog circuits is more limited. Thus, if designers can easily meet the design specs using manual design techniques, they are less likely to go through the setup effort to use an optimizer.

The Near Future: Adoption of optimizers will continue for high-performance analog design, for design migration, and for digital library creation. For optimizers to be valuable in the other areas of analog design, the optimizer must be an integral part of the design environment. The setup effort to use the optimizer must be a minimal increment on top of the existing manual design process. With FastSPICE / behavioral modeling and the appropriate hierarchical design methodology, performance optimization will find its way into use at the system level as well.

1.4 Yield Analysis and Optimization

Current Status: Performing process variation analysis, such as Monte-Carlo simulation, continues to be popular in the IDMs (Independent Device Manufacturers) where the designer has easy access to statistical process data. IDMs successfully improve their chip yields using variation analysis. Variation analysis is less popular in fabless design houses because accurate statistical process data is currently harder to come by, so process corners are typically used there. Modern variation analysis techniques not only inform the designer about how their design is going to yield but also point them to the problem devices in the circuit that are causing the yield problems.

The Near Future: Semiconductor devices sizes are shrinking, causing process variation effects to have a much larger impact on a circuit's performance and yield. Designers will increasingly use variation analysis to maintain reasonable yields on analog and high-performance digital circuits built on 90, 65 and 45 nm processes. Foundries are providing better and better statistical information, just in time for the fabless houses. To further improve yield, designers will be analyzing not only process variation effects but also layout variation effects, such as interconnect and special variation effects.

1.5 Layout Awareness, Layout Automation

Current Status: Layout-versus-schematic tools, parasitic extraction tools, and parasitic back-annotation into schematics are standard; even schematic-driven layout tools are common. For layout design itself, parameterized cells (Pcells, i.e. automated layout building blocks), are near-standard. Once a company has the Pcell infrastructure in place, layout design rules are taken into

account during the design phase with little or no extra effort on the designer's part. Tools that fully automate standard cell layout are also popular among digital standard cell library designers. These tools can automatically create predictable layouts for hundreds of digital standard cells that require little to no modification by the layout engineer. As for layout automation in analog design, partial automation such as point-to-point routing is widely used. For fully automated placement and fully automated routing, adoption is mixed; the best traction to date has been in process migration and ECO. The up-front efforts needed to configure constraints are a critical issue, though CAD work continues to reduce this problem.

The Near Future: As chip speeds increase, the tightness of information coupling between front- and back-end design will increase accordingly. We'll move from "layout-aware front-end design" and "electrically-aware layout" to broad use of a unified database at the front- and back-end that all tools use.

1.6 Analog Structural Synthesis

Current Status: Transistor-level structural synthesis, the automatic generation of a circuit topology from supplied constraints, has been a long-time dream. But there are challenges. The synthesized circuit must work across all process and environmental conditions, and must be easy to lay out. The synthesis setup effort must be much easier than designing the circuit by hand. Finally, the overwhelming challenge is that the synthesized result must ultimately be silicon-accurate, and *trustworthy*. While a few academic synthesis prototypes have shown promise on well-constrained problems, to date no scalable analog structural synthesis technique achieves these goals.

The Near (Far?) Future: For structural synthesis to be adopted, one needs to invent a synthesis solution that meets the above criterion.

1.7 Conclusion

Has analog circuit design changed in the past 30 years? Indeed it has! In addition to the algorithmic advancements discussed above, schematic and layout capture tools have certainly proven to be a large improvement over manipulating circuits in text-based formats. The shift in label from "analog" to AMS signifies a trend towards system-level design. Going forward, time to market pressures will encourage designers to try out, and adopt, new automation methodologies. To gain industry adoption, a new analog CAD tool must require minimum setup. Minimizing the setup effort means new automation tools must be easily extendable from the existing design methodologies. Similar to how new simulation techniques reuse the same netlist and device models, new environments for setting up analog automation must naturally extend from the manual design setups. Open database frameworks such as OpenAccess should help reduce the setup effort by facilitating data compatibility and reuse between tools. Lastly, for designers to adopt a new automation tool, the new tool must help them solve real design problems that are difficult to tackle with the existing design methodology.

Design Automation for Analog: The Next Generation of Tool Challenges

Rob A. Rutenbar
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, USA

rutenbar@ece.cmu.edu

ABSTRACT

The decade of the 1990s saw the first wave of practical “post-SPICE” tools for analog designs. A range of synthesis, optimization, layout and modeling techniques made their way from academic prototypes to first-generation commercial offerings. We offer some pragmatic prognostications for what the next wave might (or, more bluntly, *should*) focus on next, as pressure to improve AMS design productivity grows.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids

General Terms

Analog, Algorithms, Design, Synthesis.

Keywords

Analog, mixed-signal, integrated circuits, computer-aided design

1. INTRODUCTION

Over the last roughly half dozen years, analog design automation tools “got real” in one important sense: a range of synthesis, optimization, modeling and layout tools moved from concept demonstrations (most commonly academic) to first-generation, supported commercial offerings. We refer to these as “post-SPICE” tools; this is convenient shorthand for one unifying characteristic of these tools – the characteristic of interest in this paper – the fact that they were *not* simulation tools. To be sure, simulators saw significant advances as well in this time frame. But for the first time, we also saw some tools specifically aimed at synthesis and optimization emerge, for sizing, for centering, for layout, and so forth.

Several recent publications survey this current terrain nicely [1-3]. Based on our own experiences, with the CMU analog toolset [4-6] and its industrial progeny [7-9] we offer the following as the essential components of the current state of the art:

- **Simulation-based sizing synthesis:** these tools support circuit-level sizing, biasing, and centering. They employ global numerical optimization techniques for robustness, and network-of-workstations parallelism for speed. The key idea is full SPICE-level simulation for each solution candidate proposed during optimization. The strategy has two key virtues: it can be used for any design (i.e., any fixed topology)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD'06, November 5-9, 2006, San Jose, CA

Copyright 2006 ACM 1-59593-389-1/06/0011...\$5.00

that one can simulate; and it produces designs that pass designer-provided simulation scripts. These tools optimally reuse the verification infrastructure that all circuit designers already build for each circuit they create. And it produces “trustworthy” results, since one can immediately see that they simulate correctly, using the designer’s own simulator.

- **Optimization-based layout:** these tools replicate at device level what ASIC-level floorplanning, placement, and routing tools do at chip level. The key components are a library of generators for common device-level analog structures (e.g., analog PCELLS), and device-level placement and shape-level routing tools sensitive to analog issues such as symmetries, crosstalk and parasitic balance.

There are several examples of successes at the analog cell level (comprising roughly 10-100 devices) using these sorts of tools; Figure 1 shows one synthesis experiment from [7].

2. CHALLENGES: NEXT GENERATION

So, if we have first-generation of tools for cell-level analog designs, what’s next? Herewith, a *short* list of *big* challenges.

2.1 Integration

Neither designers, nor tools, exist in a vacuum. By “integration” we mean the process by which tools, GUIs, database schemas, usage models, etc., *co-evolve* to become maximally useful to working designers. For simulation tools, we have seen huge improvements from this co-evolution process: point tools for schematic capture, netlisting, simulation, waveform viewing, shapes-level layout, cross-probing, etc., are highly integrated today. This was certainly *not* the case when these tools appeared in the late 1980s. Similarly, the introduction of logic synthesis tools ultimately caused significant changes in the surrounding

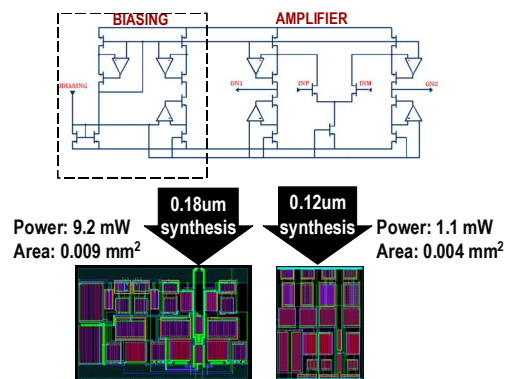


Figure 1. Example of industrial cell-level sizing/layout synthesis, from circuit experiments of [7].

RTL infrastructure; a good example is the emergence of synthesizable subsets of Verilog and VHDL.

We are still at the beginning of this co-evolution process for the post-SPICE analog tools. It is unfortunate that this process is underappreciated in academic circles, and regarded as uncreative spade-work down in the trenches of database fields and glue scripts. However, the integration process is often the make-or-break step in the path from concept to widespread adoption. A critical case in point here is the management of analog design constraints.

2.2 Constraint Extraction/Management/Reuse

Optimization-based tools have an uncanny knack for producing horrendous results when they are inappropriately set up. This is especially true for analog circuits and layouts, where even designs with a small number of elements may be subject to a large number of critical constraints. This complexity of constraints in the analog and mixed-signal world presents both challenges and opportunities.

Consider a typical analog design team, comprising a mix of circuit engineers and layout technicians. If the team has been working together for some time, and has a portfolio of successful prior designs, it has almost certainly evolved a detailed vocabulary for specifying critical topological, electrical, geometric, thermal, etc., constraints. The good news (for the team, anyway), is that such information exists. The bad news for those of us in the CAD business is that *this information is almost never written down*. Worse, when it is, its form differs from team to team, product to product, and company to company.

Extracting this information is essential for many reasons. We need it to drive our synthesis and optimization tools. (Indeed, we have lots of evidence [4-9] that when properly constrained, these tools can produce excellent, competitive designs.) We need it to parameterize error checking functions. We need it if we ever hope to make reusable IP for analog circuits. In short, we need this information to evolve a design environment with a seamless spectrum of design entry, design editing, design synthesis/optimization, design verification, and design reuse tools for the analog and mixed-signal universe.

This problem features many of the things that make CAD work really challenging: it's ill-defined, crosses abstraction boundaries (electrical, geometric, hierarchical), and needs to be parameterizable to adapt to different design styles, design groups, designed products. And, best of all, if we fail to do it right, our beautiful first-generation synthesis tools will all end up gathering dust in a corner somewhere, while our overworked analog design colleagues retreat back to manual editors and lots and lots of SPICE jobs.

Some of the work to be done is "integration" as discussed previously. We need to lower the barriers to entry of these constraints, making them a natural and expected part of the design process. This work will happen in the commercial sphere. OpenAccess [10], the open source industry-wide database initiative is a very important step in this direction.

However, there is also opportunity for longer range fundamental research. Take any complex circuit schematic/layout from a high-

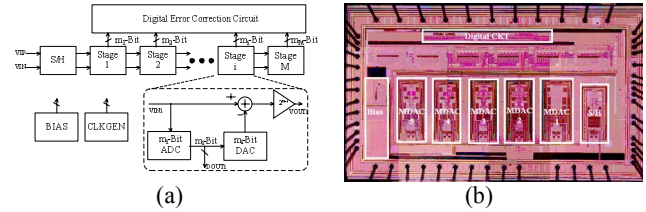


Figure 2. Hybrid AMS design using top-level analytical models with cell-level simulation-based synthesis, from [13]. (a) Overall pipelined ADC architecture. (b) Final 13b 40MS/s ADC layout, 364mW, 73.8dB SNR in 0.25um 3.3V TSMC CMOS.

performing analog team, and ask “*what’s critical about this design?*”. One will be amazed at the density of information encoded in a few essential annotations, or a small set of critical simulation waveforms. The goal is to be able extract these kinds of implicit “meta-constraints” *without* having to bother the designer. It’s a serious, exciting analog CAD challenge.

2.3 System Design/Exploration/Optimization

The emerging first generation of analog synthesis/optimization tools targets cell-level designs in the range of 10-100 devices. One significant reason is the use of simulation-based optimization, which visits many design candidates and simulates each one at full SPICE-level.

At system level, we may have 10-100 fundamental circuit *blocks*, not transistors. We cannot simulate these designs flat at the device level very efficiently. So-called “fast SPICE” engines make flattened simulation times more bearable, but still don’t support the thousands of candidate evaluations that simulation-based optimization loops rely upon. Attempts to bypass the simulation-based strategies, e.g., by using all-analytical equation-based descriptions based on convex (and thus easily optimized) descriptions (e.g., [11]), proved to be a dead end. The convex models are mathematically elegant, but too expensive to build for each new circuit, and too inaccurate versus detailed simulation.

So, what are the important tool challenges to be addressed here? Much of system level design is about trade-off analysis, understanding if a system architecture is correct, and how far it can be pushed – *before* one has fully designed it. Can we help designers make the best trade-off decisions, up at this much less concrete level of design detail? Can we help refine a design candidate to transistor level more quickly, to see if any of the components are too intractable (or too risky) to design? *These* are challenging and interesting analog optimization problems.

There are a variety of evolving approaches in this area. Hybrid schemes are one strategy worth mention. These use simulation-based synthesis engines but mix circuit level simulations for key cells with analytical formulations for top-level design tradeoffs. The work of Mukherjee et al. in [12-13] is one nice example, illustrated in Figure 2. They show how to use these ideas to explore architecture alternatives for a pipelined ADC under tight power goals, and how to synthesize components for an optimal circuit design once an optimal architecture has been selected.

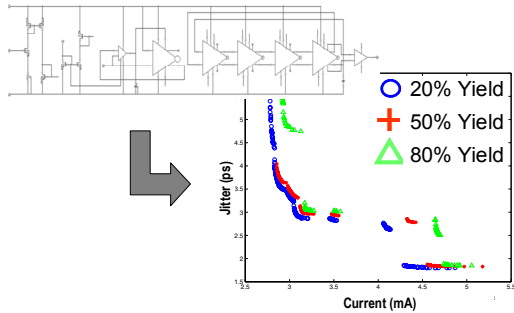


Figure 3. Statistical Pareto tradeoff curves from [16], show statistically achievable VCO current (i.e., power) vs. jitter .

Another strategy strives to extract explicit component level tradeoffs – Pareto surfaces – and use these tradeoff curves as the basis for efficient optimization at system level [14-16].

In general, this remains a rather open problem. We need flexible methods that are not only tractable, but also attractive to practicing system designers who rely almost exclusively on fast simulation. We need work not only on optimization and modeling, but also on the integration and constraint management consequences of these algorithms.

2.4 Statistical AMS Design

It should come as no surprise that designing analog, RF and mixed-signal circuits in increasingly scaled digital technologies poses new CAD challenges. The devices we will be using are increasingly subject to a wide range of both systematic and random perturbations. Their resulting performance characteristics are not only poor for many analog purposes (e.g., gain is very low) but also distributed much more widely about the nominal mean parameter values.

Most of the first-generation tools evolved to attack nominal design problems. Most of the simulation-based sizing engines can be (or, have been) married with various Monte Carlo strategies to address, at least to first order, the statistical case. However, in our opinion, we are still just scratching the surface here. We mention two specific opportunities.

One area is *circuit/layout co-design*. It is already true that the most high-performance circuits are implicitly co-designed: layout decisions, for matching, for isolation, for cross-talk, etc., are being juggled from the moment the circuit topology is defined and sizing begins. Formulating this as an explicit co-design problem seems a very attractive – and computationally challenging – problem.

Another area is *statistical system-level design*. Most synthesis experiments at system-level have targeted the nominal case, since statistical variation is difficult to capture in large scale optimization. There are many interesting strategies emerging. Our recent work at CMU [16] uses cell-level synthesis to build *statistical tradeoff curves*: tradeoff surfaces that “guarantee” that, under statistical parameter variation, some prescribed fraction (a yield level) for a given performance level. Figure 3 shows one such statistical Pareto curve. Other approaches is deterministic optimization strategies to maximize *design margins* for approximate versions of the system level problem [17], or extend *response surface methods* to high dimensional, highly correlated statistical scenarios [18]. Much remains to be done, when dealing

with system designs and at the same time, statistically varying components.

3. SUMMARY

A first generation of post-SPICE synthesis/optimization tools is currently making the transition to industrial use. We suggest four areas as priorities for next-generation work: careful integration, painless constraint extraction, practical system-level design assistance, and statistical design for circuits, layouts, and systems. These four areas comprise a *short* list of *big, open* problems in analog CAD.

REFERENCES

- [1] G. Gielen, R. Rutenbar, “Computer-Aided Design Of Analog And Mixed-Signal Integrated Circuits,” *Proceedings of the IEEE*, Vol. 88, No. 12, pp. 1825-1854, December 2000.
- [2] R. A. Rutenbar, G. G. E. Gielen, B. Antao, eds., *Computer-Aided Design of Analog Integrated Circuits and Systems*, Wiley-IEEE Press, April 2002.
- [3] J. Roychowdhury, G. Gielen, R.A. Rutenbar, “Hierarchical Modeling, Optimization and Synthesis for System-Level Analog and RF Designs,” *Proceedings of the IEEE*, to appear, 2007.
- [4] M. Krasnicki, R. Phelps, R. A. Rutenbar, L. R. Carley “MAELSTROM: Efficient Simulation-Based Synthesis for Custom Analog Cells,” in *Proc. ACM/IEEE DAC*, June 1999.
- [5] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, “A Case Study of Synthesis for Industrial-Scale Analog IP: Redesign of the Equalizer/Filter Frontend for an ADSL CODEC,” *Proc. ACM/IEEE DAC*, June 2000.
- [6] J.M. Cohn, D.J. Garrod, R.A. Rutenbar and L.R. Carley, *Analog Device-Level Layout Automation*, 285 pp., Kluwer Academic Publishers, Boston, MA, 1994. ISBN: 0-7923-9431-3.
- [7] A.H. Shah, Neoliner, S Dugalleix and F Lemery, “High-Performance CMOS-Amplifier Design Uses Front-To-Back Analog Flow,” *EDN*, 10/31/2002.
- [8] E Hennig, R. Sommer, L. Charlack, ” An Automated Approach For Sizing Complex Analog Circuits In A Simulation-Based Flow,” *Proc. Design Automation and Test, Europe (DATE)*, March 2002.
- [9] K. Oda, L. Prado, and A. J. Gadiant, “A New Methodology for Analog/Mixed-Signal (AMS) SoC Design that Enables AMS Design Reuse and Achieves Full-Custom Performance”, *Proc. ACM/IEEE Electronic Design Processes Workshop (EDP)*, April 2002.
- [10] Silicon Integration Initiative (Si2), www.si2.org.
- [11] M. Hershenson, “Design Of Pipeline Analog-To-Digital Converters Via Geometric Programming”, *Proc. ACM/IEEE ICCAD*, Nov. 2002.
- [12] Y.-T. Chien, D. Chen, J.-H. Lou, G.-K. Ma, R. A. Rutenbar, and T. Mukherjee, “Designer-Driven Topology Optimization for Pipelined Analog to Digital Converters,” in *Design Automation and Test in Europe (DATE 2005)*, March 2005, pp. 279-280.
- [13] Y.-T. Chien, L.-R. Huang, W.-T. Chen, G.-K. Ma, and T. Mukherjee, “SPEED: Synthesis of High-Performance Large Scale Analog/Mixed Signal Circuit,” in *IEEE Int’l Symp. on Technology, System and Applications, Design, Automation and Test (VLSI-TSA-DAT ’05)*, April 27-29, 2005, Hsinchu, Taiwan.
- [14] J. Zou, D. Mueller, H. Graeb, U. Schlichtmann, “A CPPLL Hierarchical Optimization Methodology Considering Jitter, Power And Locking Time,” *Proc. ACM/IEEE DAC*, July 2006, pp. 19-24.
- [15] G.G.E. Gielen, T. McConaghy, T. Eeckelaert, “Performance Space Modeling For Hierarchical Synthesis Of Analog Integrated Circuits,” *Proc. ACM/IEEE DAC*, June 2005, pp. 881-886.
- [16] S. K. Tiwary, P. K. Tiwary, R. A. Rutenbar, “Generation Of Yield-Aware Pareto Surfaces For Hierarchical Circuit Design Space Exploration,” *Proc. ACM/IEEE DAC*, July 2006, pp. 31 – 36.
- [17] X. Li, J. Wang, W. Chiang and L. Pileggi, Performance-Centering Optimization for System-Level Analog Design Exploration”, *Proceedings of the ACM/IEEE ICCAD*, November 2005.
- [18] X. Li, P. Gopalakrishnan, Y. Xu and L. Pileggi, “Robust Analog/RF Circuit Design With Projection-Based Posynomial Modeling,” *Proc. ACM/IEEE ICCAD*, pp. 855-862, 2004.

Automation in Mixed-Signal Design: Challenges and Solutions in the Wake of the Nano Era

Trent McConaghy
K.U. Leuven, ESAT-MICAS
Kasteelpark Arenberg 10
B-3001, Belgium

trent.mcconaghy@esat.kuleuven.be

Georges Gielen
K.U. Leuven, ESAT-MICAS
Kasteelpark Arenberg 10
B-3001, Belgium

georges.gielen@esat.kuleuven.be

ABSTRACT

The use of CMOS nanometer technologies at 65 nm and below will pose serious challenges on the design of mixed-signal integrated systems in the very near future. Rising design complexities, tightening time-to-market constraints, leakage power, increasing technology tolerances, and reducing supply voltages are key challenges that designers face. Novel types of devices, new process materials and new reliability issues are next on the horizon. We discuss new design methodologies and EDA tools that are being or need to be developed to address the problems of designing such mixed-signal integrated systems.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design aids –*graphics, layout, placement and routing, simulation, verification*

General Terms

Algorithms, Design, Verification.

Keywords

Analog, mixed-signal, integrated circuits, computer-aided design

1. Introduction

Shrinking device geometries leads to two overarching trends:

- More functionality is possible for the same area on a chip, which in nanometer-size geometries means *systems on chips*.
- Device sizes get increasingly closer to the size of individual atoms, so variations become increasingly important.

These trends correspond to two major challenges: handling system-level design, and handling process variations, both of which are an *increase in problem complexity*. We first discuss the problem of maintaining design insight in light of rising complexity and new processes. Then, we discuss tools and approaches that address system-level design and variation-aware design. Farther into the future, the same challenges will rise in difficulty; we discuss how structural synthesis might ultimately play a role and what other tools and methods will be needed.

2. Design Insight and Knowledge Extraction

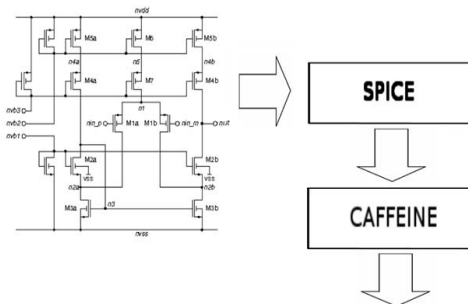
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD'06, November 5-9, 2006, San Jose, CA

Copyright 2006 ACM 1-59593-389-1/06/0011...\$5.00

For designers to remain in charge of their designs, they have to understand their circuits and the major relationships between the design variables and the circuit's performances. Yet maintaining this *insight* is becoming increasingly difficult as problem complexity rises or when new technologies and devices are being utilized. Even when CAD tools are used, these need to be set up with the proper constraints in order to generate acceptable results. Setting up the constraints requires insight into the design problem and the circuit as well.

Knowledge extraction tools are a means to *accelerate designer insight*. Enhanced insight leads to better decision-making in circuit sizing, behavioral modeling and verification, layout and topology design, regardless of the level of design automation. In this sense, knowledge extraction tools are a key way for CAD to build trust with designers, and are as such complementary to design automation tools. Symbolic analysis [1] and symbolic modeling [2][3] are examples of knowledge extraction tools. While symbolic analysis uses algorithmic methods to obtain analytic equations that characterize the circuit, symbolic modeling uses algorithms such as data mining to extract knowledge and design relations that might otherwise be "hidden" in raw data (e.g. simulation data). Because recent methods use SPICE simulation data as part of their inputs, they are general enough to cover arbitrary nonlinear circuits, technologies and analyses (e.g. transient) with good accuracy. Figure 1 illustrates this for the Caffeine tool [3] that uses genetic programming to evolve symbolic models that best fit the SPICE simulation data.



Perf.	Expression
A_{LP}	$-10.3 + 7.08e-5 / id1$ $+ 1.87 * \ln(-1.95e+9 + 1.00e+10 / (vsg1*vsg3) + 1.42e+9 *(vds2*vds5) / (vsg1*vgs2*vsg5*id2))$
I_n	$10^{(5.68 - 0.03 * vsg1 / vds2 - 55.43 * id1 + 5.63e-6 / id1)}$
PM	$90.5 + 190.6 * id1 / vsg1 + 22.2 * id2 / vds2$
V_{offset}	$-2.00e-3$
SR_p	$2.36e+7 + 1.95e+4 * id2 / id1 - 104.69 / id2 + 2.15e+9 * id2 + 4.63e+8 * id1$
SR_n	$-5.72e+7 - 2.50e+11 * (id1*id2) / vgs2 + 5.53e+6 * vds2 / vgs2 + 109.72 / id1$

Figure 1: Knowledge extraction with CAFFEINE [3] that mines SPICE simulation data to produce template-free symbolic models.

3. System-Level Design

Having a whole system on a chip leads to a whole host of challenges. But at the very core, the biggest challenge is the sheer complexity of the design. Doing the whole design “flat” is intractable, so hierarchical decomposition is necessary. So then, exactly *how* does one traverse the hierarchy? The top-down constraint driven methodology (TDCD) (e.g. [4]) is often assumed to be the “target methodology”, as in the MEDEA+ EDA roadmap [5]. But it has been tough to do in analog because of the issue of knowing what combinations of specifications are feasible. Two main alternatives have emerged in the literature:

1. Doing bottom-up feasibility modeling as a precursor step to top-down constraint driven [6][7]
2. Multi-objective bottom-up design (MOBU) [8][9][11].

Each has its own way of handling constraints and objectives. Both enable one to get an optimal system-level trade-off of possible combinations of specifications. Even after one has selected a hierarchical traversal approach, several issues remain, such as what method to use to estimate performances at all levels (e.g. using SPICE or behavioral simulators, regression-based models, etc.), how to handle interactions and mutual constraints between different subblocks, how to account for process variations, etc.

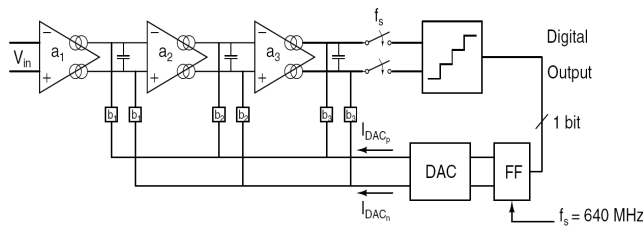


Figure 2: High-speed $\Delta\Sigma$ A/D modulator.

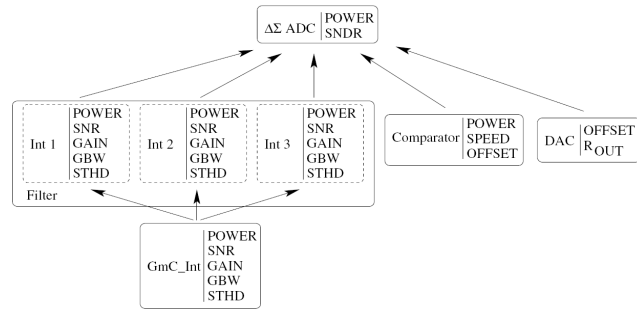


Figure 3: MOBU-style hierarchical traversal propagates trade-offs in a bottom-up fashion, maintaining all the sized designs along the way.

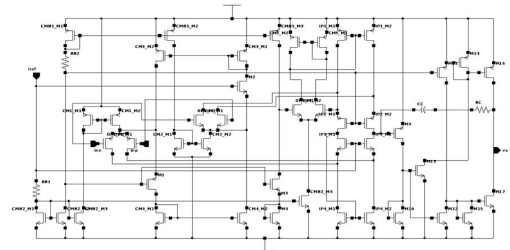
To give a flavor of the state of the art, MOBU was recently demonstrated in the design of a high-speed $\Delta\Sigma$ A/D modulator for a WLAN 802.11a/b/g standard [9]. The system-level topology is shown in Figure 2. The hierarchical traversal proceeded as follows. The design was decomposed into seven separate subblocks, as shown in Figure 3. Then, at the very bottom block (GmC_Int), a multi-objective optimization was run, generating a set of designs, each trading off some specs for others [10]. At the Filter block, there were three integrators, each of which could be any of the designs created in the GmC_Int optimization. Three multi-objective optimizations generated trade-offs for the Filter, Comparator, and DAC respectively. Finally, a multi-objective optimization was run at the ADC level, using the trade-offs from

the Filter, Comparator and DAC as part of its design space. The total runtime was about 3 days; in comparison the manual reference design took 6 months. Moreover, MOBU generated better designs; for example, one design had approximately the same performances but *half* the power consumption of the manual design [9].

4. Variation-Aware Design

Designers not only have to deal with ever-larger designs, they also have to use ever-smaller devices. With that comes a continual increase in process variations, posing a threat to yield. The MEDEA+ roadmap [5] talks of yields of 95% at 0.35 μm , which reduce to just 50% at 90 nm. Variation-aware design needs good statistical modeling; unfortunately, such models have 9 to 15 or more random variables *per device* in the circuit [12]. As a result, the designer has to manage an unreasonable number of variables. The designer’s rules of thumb can disappear too; for example the common tactic of merely increasing area to reduce mismatch may actually not help much because of the nonlinear relation between W and L and mismatch in newer processes (Figure 6 in [12]).

While Monte-Carlo sampling is a good first step to variation-aware design, it doesn’t answer critical design questions such as identifying how design variables interact and affect yield. Caffeine [3] can address this: Figure 4 shows a Caffeine-generated equation for Cpk (“process capability”) of a 50-transistor amp having 68 design variables, which could subsequently be used for manual or automatic yield optimization.



$$\begin{aligned}
 \text{Cpk} = & 1231.4 \\
 & - 9.39\text{e}+08 * \text{dp2_w}^2 * \sqrt{\text{dp1_w}} * \min(6.60\text{e}+07 - 76.9 / \sqrt{\text{Cc}}, 0.104) \\
 & + 1.21\text{e}+12 / \min(1\text{e}+10 - 2.48\text{e}-05 / (\sqrt{\text{dp2_w}} * \text{Cc}), -4.96\text{e}+06) \\
 & - 0.0012 / \sqrt{\text{Cc}} + 4.21\text{e}+06 * \text{mt4_w}^2 / \text{mt1_w}
 \end{aligned}$$

Figure 4: Amplifier, and an equation for its Cpk that was generated by the Caffeine tool [3]. $\text{rmse}_{\text{test}} = 6.3\%$.

As process scaling continues more flows and tools must be variation-aware. At first this may be by simply adding safety margins or post-design yield tuning; but ultimately all the design flows and tools will have to be statistically-centric from the start.

5. Topology Design

Moore’s Law will continue to march on, so big systems will get even bigger and process variations will have even more variation. Gates will be leakier, and materials will continue to change. In addition, supply voltages will be scaling, reducing the headroom for analog circuits. All this has implications for circuit topology design, since existing topologies may not function anymore. Also, gates are becoming virtually free, giving opportunities to use digital logic to construct or calibrate analog behavior. This all leads to challenges (opportunities) for topology design, manual *and* automatic. Two approaches for automated topology generation can be distinguished:

Approach 1. In the “building-block substitution” approach, an optimal topology is found by iteratively substituting different

subblock implementations in some predefined topology template. This could be done manually where the designer goes from abstract specification down to detailed design, or could be driven by the specification in an optimization loop like in Darwin [13]. An example of recent results in ADC design [14] is in Figure 5 (left), which shows example operators that drill down towards implementation. By putting those sorts of operators into a multi-objective optimization framework, a whole set of trustworthy topologies can be generated, as Figure 5 (right) illustrates.

Approach 2. Structural synthesis that generates truly novel designs, driven by the specifications. At the cell level, the closest we have seen is [15], which used genetic programming to automatically reinvent some small circuits from around-2000 patents, a pretty impressive feat. Unfortunately, the computational resources needed to generate those were incredibly high, and if the circuit problems were modeled with more industrial constraints, then the runtime would be about 150 years on a 1000-node 1-GHz cluster [16].

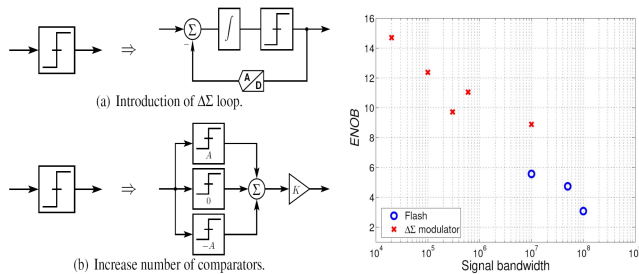


Figure 5: Left: Example “topology refinement” operators. Right: System-level performance trade-offs; the tool automatically selects the most promising topology for each combination of specifications.

In our opinion, *trust* is an even bigger issue than computational cost. Trust issues occur whenever previously-unseen substructures are generated by the synthesis tool. For designers to really trust a novel circuit, they need to see it working in silicon. Structural circuit synthesis can be helpful to accelerate development of new design techniques in new processes, new devices, etc. for automated *or* manual flows. This use case has actually been happening in other domains such as quantum circuit design [17] and multi-valued logic design [18].

The MEDEA+ roadmap targets “analog [structural] synthesis” for 2009-2010. AMS CAD will certainly notch up a level of excitement when this dream becomes reality.

6. Conclusions

Moore’s Law will keep charging, leading to increased complexities at the very top (system-level design) and at the very bottom (variations gone crazy). We have discussed CAD research that addresses hierarchical system-level design and variation-aware design. We have emphasized the importance for the designer to maintain insight into the circuit and how tools can help in knowledge extraction. Going farther into the future, we expect a need for more tools in signal integrity analysis (such as EMC analysis), for more automated model generation, for more accurate modeling at higher operating frequencies and for true structural synthesis at both circuit and system levels.

ACKNOWLEDGEMENTS

Funding for the reported research results is acknowledged from IWT/Medea+ Uppermost, Solido Design Automation and FWO Flanders.

REFERENCES

- [1] G. Gielen, “Techniques and Applications of Symbolic Analysis for Analog Integrated Circuits: A Tutorial Overview”, in *Computer Aided Design of Analog Integrated Circuits And Systems*, R.A. Rutenbar et al., eds., IEEE, pp. 245-261, 2002.
- [2] W. Daems, et al. "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE Trans. CAD* 22(5), May 2003, pp. 517-534.
- [3] T. McConaghy, et al., “CAFFEINE: Template-Free Symbolic Model Generation of Analog Circuits via Canonical Form Functions and Genetic Programming”, *Proc. DATE 2005*.
- [4] H. Chang, et al. *A Top-Down, Constraint-Driven Design Methodology for Analog Integrated Circuits*. Kluwer Academic Publishers, 1997.
- [5] MEDEA+ EDA Roadmap, Version 5, 2005, <http://www.medeaplus.org>
- [6] F. De Bernardinis, et al, “Support Vector Machines for Analog Circuit Performance Representation,” *Proc. DAC 05*
- [7] D. Mueller, et al. “Deterministic Approaches to Analog Performance Space Exploration”, *Proc. DAC 2005*, pp. 869-874, 2005.
- [8] T. Eeckelaert, et al. “Efficient Multiobjective Synthesis of Analog Circuits using Hierarchical Pareto-optimal Performance Hypersurfaces,” *Proc. DATE*, 2005.
- [9] T. Eeckelaert, et al. “Hierarchical Bottom-up Analog Optimization Methodology Validated by a Delta-Sigma A/D Converter Design for the 802.11 a/b/g Standard,” *Proc. DAC*, pp. 25-30, 2006.
- [10] B. De Smedt, G. Gielen, “WATSON: Design Space Boundary Exploration and Model Generation for Analog and RF IC Design,” *IEEE TCAD* 22(2), Feb. 2003, pp. 213–224.
- [11] S. Tiwary, P. Tiwary, R. Rutenbar, “Generation of Yield-Aware Pareto Surfaces for Hierarchical Circuit Design Space Exploration,” *Proc. DAC 2006*, pp. 31-56, 2006.
- [12] P. Drennan et al. “Understanding MOSFET Mismatch for Analog Design,” *IEEE JSSC*, Vol. 38, pp.450-456, 2003.
- [13] W. Kruiskamp, et al., “DARWIN: CMOS Opamp Synthesis by means of a Genetic Algorithm”, *Proc. DAC*, 1995.
- [14] E. Martens, G. Gielen, “Top-Down Heterogenous Synthesis of Analog and Mixed-Signal Systems,” *Proc. DATE*, 2006.
- [15] J.R. Koza, et al., *Genetic Programming IV*, Kluwer, 2003.
- [16] T. McConaghy, G. Gielen, “Genetic Programming in Industrial Analog CAD: Applications and Challenges”, *Genetic Programming Theory and Practice III*, ch. 19, pp. 291-306, 2005.
- [17] L. Spector, H. Bernstein, “Communication Capacities of Some Quantum Gates, Discovered in Part through Genetic Programming,” In *Proc. Intl. Conf. on Quantum Comm., Measurement., and Computing*, pp. 500-503, 2003.
- [18] J. Miller, et al., “The Genetic Algorithm as a Discovery Engine: Strange Circuits and New Principles,” in *Proc. AISB Symp. on Creative Evolutionary Systems*, UK, 1999.