Use of Symbolic Performance Models in Layout-Inclusive RF Low Noise Amplifier Synthesis

Mukesh Ranjan[†], Amitava Bhaduri[†], Wim Verhaegen[§], Bhaskar Mukherjee[‡],
Ranga Vemuri[†], Georges Gielen[§], Andrea Pacelli[‡]

[†]Dept. of ECECS, University of Cincinnati, Cincinnati, OH-45221 USA.
[§]Dept. of EE, ESAT-MICAS, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium.
[‡]Dept. of ECE, SUNY Stony Brook, Stony Brook, NY-11974 USA.

Abstract— **In this paper we present a layout-in-loop synthesis method for radio-frequency LNAs, which uses symbolic performance models (SPMs), parameterized layout generator and high-frequency extraction techniques in the synthesis loop. The primary focus of this work is on performance estimation using efficient SPMs and development of techniques to include layout parasitics symbolically into the SPMs before the start of synthesis. SPMs for noise figure and distortion parameters are obtained using repetitive and weakly nonlinear symbolic analysis and are stored as pre-compiled Element Coefficient Diagrams (ECDs). Speedy layout generation is achieved by using parameterized procedural layout generators and full parasitic extraction is done by using multiple extractors. Quasi-static extraction is used to obtain the critical parasitic effects of interconnects and on-chip inductors. The proposed methodology is used for the synthesis of Low Noise Amplifiers (LNAs).**

I. Introduction & Related Work

Circuit synthesis is the process of determining the numerical values for unsized circuit elements of a fixed circuit topology, by using a combinatorial optimization algorithm, in order to satisfy a set of performance constraints [6]. This traditional process when applied to radio-frequency (RF) circuits, suffers from two significant shortcomings. First, the parasitics introduced in the layout phase are not considered during optimization. As the performance of an RF circuit is extremely sensitive to these parasitics, it often leads to the failure of the sized circuit after layout. The second drawback is the extremely expensive computations for performance estimation because of using numerical analysis solely in the synthesis loop.

The problem of performance degradation due to layout parasitics can be alleviated by either layout-aware or layout-in-loop circuit synthesis. While layout-aware techniques offer fast timing closure, the parasitics are difficult, and often impossible, to estimate accurately. In layout-in-loop synthesis, layout generation and extraction is done within the synthesis loop with the hope of capturing the parasitic effects accurately. Several attempts have been made to synthesize RF circuits using layout-aware and layout-in-loop techniques, but none of them consider the full extraction of circuits including the metallic interconnects [4], [5], [11], [12].

The problem of computationally intensive numerical simulations can be avoided to an extent by using circuit performance models for estimating the behavior of a particular design. The generation and evaluation of accurate symbolic performance models (SPMs) is fast [13] and therefore is the method of our choice. We propose a method which combines parameterized layout generators, rule-based extractors, dedicated extractors for on-chip inductors and interconnects and SPMs for fast performance closure during analog circuit sizing. A similar approach was proposed by us in [10] but it is limited to linear analog circuit synthesis and the parasitic-inclusion techniques are relevant only to rule-based extractors. This paper presents a synthesis methodology for nonlinear RF LNA circuits along with techniques for the inclusion of high-frequency quasi-static layout effects into the SPMs.

The rest of the paper is organized as follows. The proposed method is described in Section 2 and techniques to include layout effects in the SPMs in Section 3. Experimental results are presented in Section 4 and conclusions in Section 5.

II. Proposed RF Circuit Synthesis Method

The proposed RF circuit synthesis environment is shown in Figure 1. The Module Specification Language(MSL) system [8] is used for layout generation. This is a language-based layout generation system, which produces a parameterized layout generator from the placement and/or routing description of any circuit under consideration. The layout generator thus obtained is instantiated in every synthesis iteration using the devices sizes furnished by the optimization engine. Since only an instantiation step is involved the layouts are generated very fast. The core of the optimization engine is the simulated annealing algorithm.

The layout instantiation step is followed a multi-way extraction. In order to extract the high-frequency parasitics associated with the interconnects as well as on-chip inductors, relevant components of the layout are passed onto a quasi-static extractor. In a parallel step, the parasitic capacitances associated with interconnects, on-chip inductors and on-chip capacitors are extracted using rule-based extractors. Transistors and their parasitics are also extracted using rule-based extractor albeit separately. This extraction step is followed by a netlist post-processing step where the netlist of extracted parasitics is sorted and compacted. A map of parasitic values which is eventually used in the evaluation of symbolic performance models is also generated in this step.

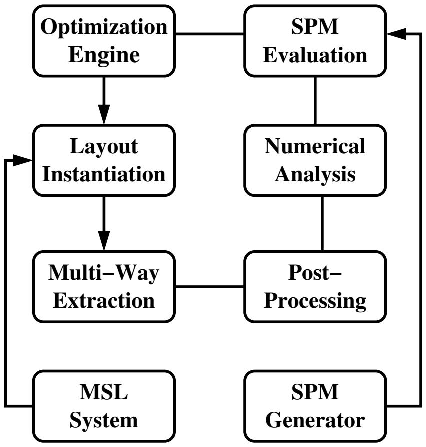

Fig. 1. Proposed Approach

In our method symbolic performance models (SPMs) are used to judge if the devices sizes proposed by the optimization engine meets the desired specifications. These models are generated only once before the start of synthesis (similar to the parameterized layout generator) and are repetitively evaluated during synthesis. During the evaluation step, SPMs use small-signal values for active devices and a set of non-linearity coeffcients. These are generated during the numerical analysis step shown in the figure. Once the performance estimates have been obtained the optimization engine (if required) proposes a new set of circuit parameter values (device sizes). These values are again fed to the layout generator to instantiate a physical layout. The process continues till the optimization algorithm converges to a design which is near optimum and satisfies all the performance constraints. The detailed description of each step follows.

A. Layout Generation and Instantiation

The MSL system [8] is used to generate parameterized layouts. MSL contains construct for hierarchical instantiation and relative placement of modules and for defining parameterized nets for routing. When compiled an MSL program yields an executable layout generator which can be quickly elaborated into a concrete layout, when concrete size information is available from the optimization engine. A library of parameterized module generators, including fingered transistors, differential pairs, current mirrors, resistors, capacitors,inductors etc, has been developed in MSL.

B. Multi-way Layout Extraction

A multi-way layout extraction, for LNA circuits is done by strategically avoiding replication or aliasing of node and by preserving the continuity of the resultant netlist [1]. The final netlist is obtained by merging all the netlists generated by rule-based and quasi-static extractors.

Rule-based Extraction: The rule-based extractor is used to extract the parasitics of active devices and the capacitances of on-chip inductors, capacitors and interconnects. First the active devices (transistors) are extracted by using a rule-based methodology. The result is a netlist file which contains the diffusion capacitances of the fingered transistors and the parasitic capacitances of the intra-modular interconnects. Then the intrinsic capacitance of the on-chip MiM capacitors are extracted by the rule-based method along with the parasitic capacitances of on-chip inductors, interconnects skeleton and poly-resistors. This produces another netlist file where the extracted capacitances are lumped at each node. The rule-based extractor used in our experimentation is the standard MAGIC extractor.

Quasi-Static Extraction: The layout devoid of active devices, MiM capacitors and poly-resistors is passed through a quasi-static extractor. This tool extracts the resistances and inductances of the metallic interconnects and on-chip inductors and outputs another netlist. The extractor used here is part of the VPEC software suite [3] for inductance extraction and sparsification. An efficient inductance calculator, based on the quasi-static limit which usually matches FastHenry results within a few percent accuracy is used . In this work we did not account for skin effect in the wires, because the cross section of the wires is smaller than the skin depth at the frequencies under consideration. VPEC does not extract the capacitance of the interconnects or the on-chip inductors, but this deficiency is obviated by merging the extracted netlist with the result of the rule-based extraction described above.

C. Post Processing of Extracted Netlists

In order to be amenable to symbolic methods, the extracted netlists have to be compacted, especially after VPEC extraction. Internally VPEC uses a program called *xray* to extract the wireframe representation of the layout from its flat version. The wireframes in essence represent the geometry of a layout and are eventually used by the VPEC extractor to obtain the parasitic netlist. Each segment in the wireframe is extracted as an Voltage-Resistance-Inductance (VRL) segment. Since, *xray* supports three-dimensional (multilevel) interconnect, each level is extracted separately but is connected through vertical wires (representing vias and contacts) to other segments. The vias and contacts are also extracted and appear in the final netlist. Due to the thorough extraction technique used by VPEC, the number of parasitic elements extracted explode in number. Conceptually it becomes difficult to capture the complete set of parasitics generated in all iterations, before the generation of SPMs. Moreover, symbolic analysis, the cornerstone of symbolic performance modeling, is incapable of handling such huge circuits. In case of the Low Noise Amplifier shown in figure 6, the five interconnects are extracted as 96 VRL segments. Such a scenario makes it necessary to post process the extracted parasitic netlist.

The first step, is sorting the netlist. In the netlist generated by VPEC, the different segments of a net are spread all over and are not bunched togther. For proper extraction of parasitic values it is essential that the segments be sorted. The net segments are therefore sorted according to their net affiliations. This is done by scanning the terminal nodes of each net segment and linking it with an earlier segment. Each net is thus stored as an acyclic graph, with vertices representing the nodes and the edges the impedance of a segment. This graphical representation of a net is then compacted by merging circuit parasitics which are in series with each other. This entails removal of a vertex with only one edge originating from it and connecting its *child* vertex to its *parent* vertex, while lumping the vertex's impedance with that of the child vertex. Due to this representation it becomes easy to generate the map of parasitic values used by SPMs and also to generate the complete compacted netlist for later simulations.

D. High-Frequency Symbolic Performance Models

Symbolic performance models (SPMs) are symbolic equations in terms of circuit parameters that represent the performance metrics of an analog circuit. An SPM is built using one or more symbolic transfer functions and a symbolic mathematical formula (performance model) which represents the relationship between a performance characteristic and the transfer functions. In circuit synthesis, SPMs are used for repetitive performance estimation during the optimization iterations. Unlike numerical simulators, analysis is not done in every iteration and only evaluation of the SPMs is needed. This results in a significant speedup of the performance estimation time. At radio-frequencies the circuit is very susceptible to noise and distortion, therefore the following performance characteristics are optimized: gain, noise figure and the distortion parameters (HD_2 & HD_3). The following subsections describe the process of obtaining noise figure and distortion SPMs and process of symbolic analysis.

1) Noise Figure: The generation of Noise Figure SPM is similar to the one attempted in [14], but our approach extends this technique to the modeling of noise figure rather than just spectral noise density. Moreover, instead of using a DDD approach with evaluation in memory we have used pre-compiled ECDs to represent the transfer functions.

2) Distortion Parameters: The symbolic formulae for distortion and intermodulation products of RF circuits are obtained through a Volterra-series based method [16]. In short, the behavior of the nonlinear circuit elements is described using a multi-dimensional Taylor series, which is truncated after the 3rd order terms. While in general this truncation is not necessarily sufficiently accurate, for the application towards RF circuits this poses no problems. Using the further assumption that higher order terms are negligible in the computation of lower order ones - which is also fair to assume for RF circuits - the circuit can effectively be composed in a number of circuits, each of which models the behavior at a frequency which is a linear combination of the input frequencies.

The construction of this models using the simplifying assumptions does not need to be repeated for each new circuit. An analysis of these effects using the general Kirchhoff laws allows us to derive a number of stamps that model the behavior of (trans)conductances and capacitances for an effect of a given order. An incremental substitution of the circuit elements by these stamps leads to a set of models in which only the linear interactions between the terminals of the nonlinear elements still need to be computed. These linear transfer functions are obtained as described above, and modeled as ECDs.

The SPM resulting from the Volterra based analysis consists of a postprocessing function combining the components at appropriate frequencies of a number of in- and output voltages and/or currents, which themselves are modeled using a hierarchy of expressions containing linear transfer functions (modeled using ECDs) and nonlinearity coefficients only. Just as is the case with the evaluation of the ECDs these SPMs depend on the topology only and need to be constructed only once. The evaluation of the SPMs additionally requires numerical values for the nonlinearity coefficients, which can be obtained using either numerical postprocessing of a operating point simulation, or through models. Since no accurate model equations are available to us, we have opted for the numerical nonlinearity calculation.

3) Symbolic Analysis and Pre-Compiled ECDs: The core of this SPM generation process is *symbolic analysis*. Symbolic analysis is a formal technique used to obtain network transfer functions in terms of symbolic circuit parameters and independent variables like frequency [7]. In circuit synthesis it is essential that the SPMs are not approximate, require minimum space, are fast to evaluate and are stored in the *s-polynomial* format [13]. In this paper we use Element-Coefficient Diagrams (ECDs) to represent a symbolic transfer function. The process of ECD-based symbolic analysis is described in [15]. In our work the ECDs are converted to C++ code and then compiled. The use of pre-compiled ECDs reduces the SPM evaluation time considerably. An significant average speedup (by a factor of 30), with respect to evaluation of ECDs stored in memory, is observed for for the transfer functions of the benchmark circuit.

III. Inclusion of Layout Effects in SPMs

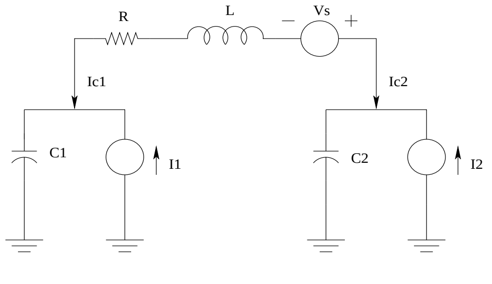

Fig. 2. PEEC Model

The perturbation of component sizes during synthesis causes variations in the layout geometry between iterations. These variations may generate varying sets of parasitic elements (resistances, capacitances and inductances) in each iteration. The variation of width of a transistor module also causes the number of fingers to change between iterations. In this section we talk about techniques to model these variations in the original circuit topology, which in turn generates *parasitic-inclusive* SPMs.

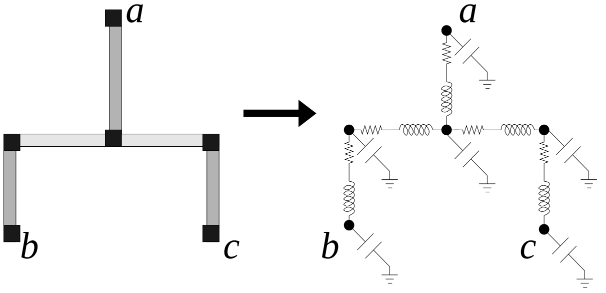

Fig. 3. Illustration of parasitic extraction

The inclusion of parasitic capacitances for active components obtained by rule based extraction can be done using analysis-based inclusion technique proposed by us in [10]. The set of parasitics generated for every segment of an interconnect is in the form of the PEEC Model [2] shown in Figure 2. The extracted circuit for a three terminal net is shown in Figure 3. Note that the voltage and current sources of the PEEC Model are not included in the circuit as the evaluation of SPMs is independent of the process of biasing a circuit.

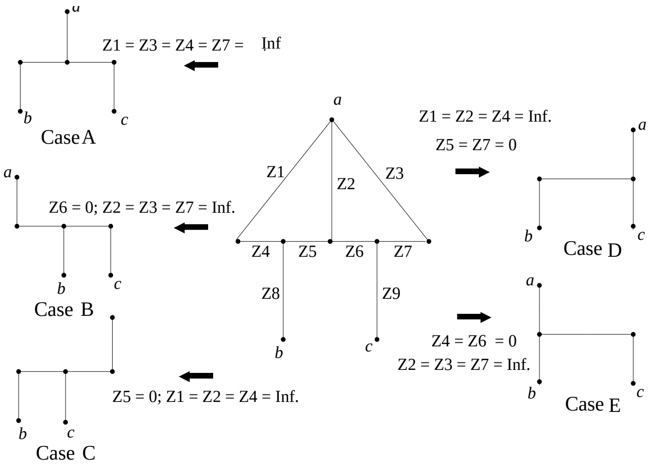

Fig. 4. Illustration of parasitic inclusion in SPMs

For multi-terminal nets the number of parasitic elements extracted in different iterations may not be the same. This is because the pin positions change and this causes the nets geometry to change. Figure 4 shows a multi-terminal net with one terminal on the top row and two terminals on the bottom row and the five possible net geometries. Based on the relative *x*-coordinate of node a, *a(x)* with respect to *b(x)* and *c(x)*, the number and position of parasitics extracted vary. To model these variations a complete parasitic model, which is a superset of all possible combinations (shown in the center of Figure 4), is included in the original circuit description which is eventually used to generate the SPMs. The actual evaluation of the SPMs is done by determining the relative coordinates of the critical nodes (a,b,c in this example) and setting the value of some resistances and conductances to zero. Figure 5 shows another example of a parasitic model for a net with two upper terminals and three lower terminals. This parasitic modeling technique can easily be extended to nets with *m* terminals on the top row and *n* terminals at the bottom row.

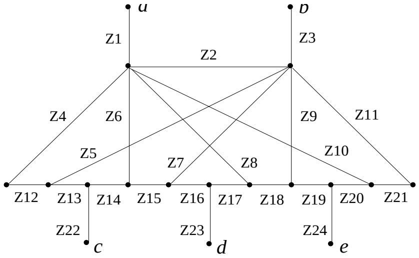

Fig. 5. Larger example of a parasitic model

IV. Experimental Results

A. Setup

The LNA in Figure 6 is chosen to demonstrate the proposed RF circuit synthesis method. In this circuit the input & outputs are matched at 50Ω and the width of transistors M0 & M1 are equal. The equal widths enable the merging of the drain of M0 with the source of M1, which in turn reduces the internal noise of the cascode transistor M1 and facilitates input matching [9]. The width of transistor M2 (which forms a current mirror with M1) is chosen to be one-tenth of the width of M1. This minimizes the power consumption in the bias circuit [9]. The large input blocking capacitance is off-chip and its value is fixed at 10pF. The SA-based optimizer explores the search space for these seven design variables: M1.W, L0, L1, L2, R0, R1 and C0. The TSMC 0.18μ technology is used and all simulations are run on SunBlade1000 with 2GB RAM.

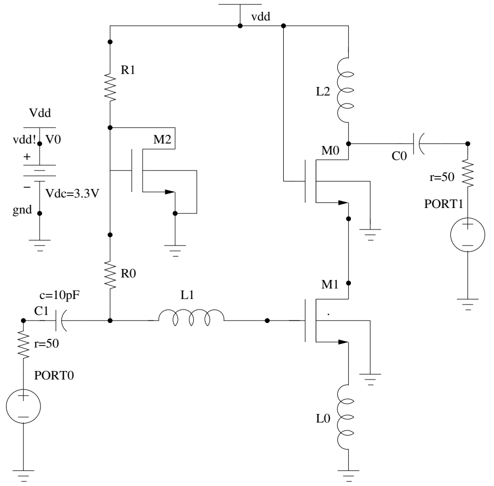

Fig. 6. Low Noise Amplifier Topology

B. Results and Discussion

The SPM generation process for the circuit described above took around 45 minutes, out of which the ECD generation time was 16 minutes 34 seconds and ECD compilation time

TABLE I

SYNTHESIS RESULTS FOR LOW-NOISE AMPLIFIER

Attribute	Constraints	Obtained	Verified	Error
Gain	≥15 dB	16.565 dB	16.548	0.102 %
NF	≤3 dB	2.754 dB	2.90	5.03 %
$H2$	≤-25 dB	-26.239 dB	-	-
$H3$	≤-20 dB	-20.782 dB	-	-

was 26 minutes 23 seconds. The number of ECDs required for *H2* are 9; for *H3* are 31 and for *noise figure* are 8. So a total of 49 ECDs were generated and compiled (including 1 ECD for *gain*). The single-ended LNA was synthesized for gain, H2, H3 and noise figure at a frequecy of 2.1 GHz. The results are presented in table III. When compared to a numerical simulator the accuracy for gain was within 0.1%, for noise figure within 5%. The slight inaccuracy in noise figure arises because we do not consider all sources of noise but only the important ones, the thermal noise due to the transistors' channel resistances and circuit resistances. The accuracy of H2 & H3 is discussed in details in [16]. The inaccuracy in H2 & H3 arises from two sources. First, is due to the fundamental assumptions associated with weakly non-linear symbolic analysis. The other source of inaccuracy is due to the use of non-linearity coeffcients which are obtained by interpolation the operating point results. The accuracy can be improved by using a tighter grid but of course that would slow down the process significantly. The total synthesis time was 13 hours. Each iteration time is 26.04s, out of which layout generation time is 1.1s, layout extraction time is 8.67s and performance estimation time is 16.23s. Figure 7 shows a layout obtained after synthesis.

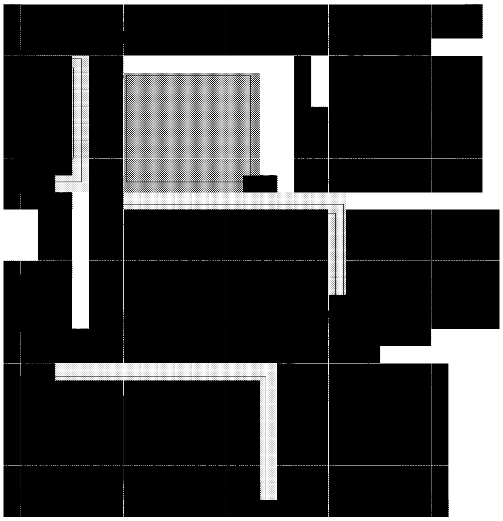

Fig. 7. Layout of the Single-ended LNA

V. CONCLUSIONS

The paper has presented a new method for RF LNA synthesis using quasi-static extraction and symbolic performance models. The novelty of the approach is the use of ECD-based symbolic performance models for RF layout-in-loop synthesis. The synthesis method can be improved significantly in terms of speed by calculating non-linearity coefficients using biasing voltages and model parameters of active components instead of numerical interpolations. This is a part of the future work along with the development of symbolic performance models for *IM3*.

ACKNOWLEDGEMENT

This work has been sponsored in part by DARPA/MTO NeoCAD program under grant number F333615-01-C-1977 monitored by the Air Force Research Laboratory.

REFERENCES

[1] A. Bhadhuri et al., “”Parasitic-Aware Synthesis of RF LNA Circuits considering Quasi-Static Extraction of Inductors and Interconnects”,” in *Proc. of MWSCAS (Accepted to appear)*, July 2004.

[2] A. E. Ruehli and A. C. Cangellaris, “”Progress in the Methodologies for the Electrical Modeling of Interconnects and Electronic Packages,” in *Proc. IEEE*, May 2001, pp. 740–771.

[3] A. Pacelli, “”A local circuit topology for inductive parasitics”,” in *Proc. of ICCAD*, Nov. 2002, pp. 208–214.

[4] C. De Ranter et al., “”CYCLONE: Automated Design and Layout of RF LC-Oscillators”,” in *Proc. of DAC*, June 2000, pp. 11–14.

[5] M. del Mar Hershenson et al., “”Design and optimization of LC oscillators”,” in *Proc. of ICCAD*, 1999.

[6] G. Gielen and R. Rutenbar, “”Computer-Aided Design of Analog and mixed-signal integrated circuits”,” *Proc. IEEE*, vol. 88, no. 12, pp. 1825–1854, Dec. 2000.

[7] G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Kluwer Academic Publishers, 1991.

[8] H. Sampath and R. Vemuri, “”MSL: A High-Level Language for Parameterized Analog and Mixed-Signal Layout Generators”,” in *Proc. of IFIP 12th Intl. Conf. on VLSI*, 2003.

[9] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge University Press, 2002.

[10] M.Ranjan et al., “”Fast, Layout-Inclusive Analog Circuit Synthesis using pre-compiled parasitic-aware symbolic performance models”,” in *Proc. of DATE*, Feb 2004, pp. 604–609.

[11] P. Vancorenland, G. Van der Plas, M. Steyaert, G. Gielen and W. Sansen, “”A Layout-aware Synthesis Methodology for RF Circuits”,” in *Proc. of ICCAD*, 2001.

[12] R. Gupta, B. M. Ballweber and D. J. Allstot, “”Design and Optimization of CMOS RF Power Amplifiers”,” in *IEEE JSSC*, 2001, pp. 166–175.

[13] C.-J. R. Shi and X.-D. Tan, “”Compact Representation and Efficient Generation of s-Expanded Symbolic Network Functions for Computer-Aided Analog Circuit Design”,” *IEEE Trans. on CAD of Integrated Circuits and System*, vol. 20, no. 7, pp. 813–827, July 2001.

[14] X.-D. Tan and C.-J. R. Shi, “”Symbolic circuit-noise analysis and modeling with determinant decision diagrams”,” in *Proc. of ASP-DAC*, Yokohama, Japan, Jan. 25–28, 2000, pp. 283–287.

[15] W. Verhaegen and G. Gielen, “”Efficient DDD-based Symbolic Analysis of Large Linear Analog Circuits”,” in *Proc. of DAC*, 2001, pp. 139–144.

[16] P. Wambacq, G. Gielen, P. Kinget, and W.Sansen, “”High-frequency distortion analysis of analog integrated circuits”,” *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 3, pp. 335–345, Mar. 1999.