An Approach to Topology Synthesis of Analog Circuits Using Hierarchical Blocks and Symbolic Analysis *

Xiaoying Wang Lars Hedrich
Department of Computer Science, University of Frankfurt
wang@em.informatik.uni-frankfurt.de
hedrich@em.informatik.uni-frankfurt.de

Abstract— This paper presents a method of design automation for analog circuits, focusing on topology generation and quick performance evaluation. First we describe mechanisms to generate circuit topologies with hierarchical blocks. Those blocks are specialized by adding terminal information. The connection between blocks is in compliance with a set of synthesis rules, which are extracted from typical schematics in the literature. Symbolic analysis has been used to select an appropriate topology quickly and to help the designer gain a better understanding of a circuit's behavior. Finally, experimental results show the creativity and efficiency of our method.

I. Introduction

Chips based on future nanotechnologies make it possible that more and more complex systems can be integrated on one single chip, which includes digital, analog and mixed-signal sections. Although most functions of such systems are realized in a digital/mixed-signal section, the analog circuits cannot be neglected, because of fundamental necessity (e.g. interface with the outside world).

EDA tools for analog design compared to those for digital design are not yet fully mature and still in the process of exploration and development, because analog design is mostly heuristic and knowledge-intensive. More and more circuits with new topology are required to fit the growing complexity of systems. Hence, analog circuit synthesis is one critical step, which includes topology generation/selection and circuit sizing [1]. Up to now there are several approaches providing ideas and solutions for automated topology synthesis of analog circuits in different ways: OASYS [2], BLADES [3] are based on a library with a set of well-defined topologies, from which a proper topology will be selected. Some sized topology generation approaches e.g. [4] and [5] use genetic algorithms to find a lot of variety of circuits topologies. Another approach from Klumperink [6] explores all elementary circuits with one or two Voltage Controlled Current Sources (VCCS).

*This work is supported by BMBF/edacentrum project SAMS under No. 01M3070D.

In this paper we present a new method in topology synthesis that can be roughly divided into two lines of work, generation and selection of circuit topology:

- Hierarchical design methodology for generation of circuit topology
  - A set of well defined blocks, such as blocks for current mirrors or for differential pair, with specialized signal information of terminals.
  - A set of extracted rules for combination between blocks in consideration of the signal information of terminals.
- Symbolic analysis methodology for selection of circuit topology
  - The circuit topologies performances are analyzed with linear symbolic analysis methods.
  - The symbolic formulas enable fast but reliable estimation of circuit performances taken matching, and the dependencies of linear device parameters in terms of size and biasing into account.

The paper is organized as follows. Section II describes the hierarchical topology generation system. Section III focuses on the symbolic analysis. Section IV presents experimental results, and finally, in section V conclusions are drawn.

II. Hierarchical Topology Synthesis

It is known that analog circuit topologies can be represented in many small subcircuits, e.g. differential pair, current mirror, cascode stages, etc. [2][7]. The designer can catch the functions of circuits more easily with help from identification of small subcircuits. The basic idea of our method is: these subcircuits are just like "building blocks" of analog circuits and our aim is how to get analog circuits using these building blocks. To do this, we have defined new blocks with terminal information for building blocks and a set of synthesis rules,

0-7803-9451-8/06/$20.00

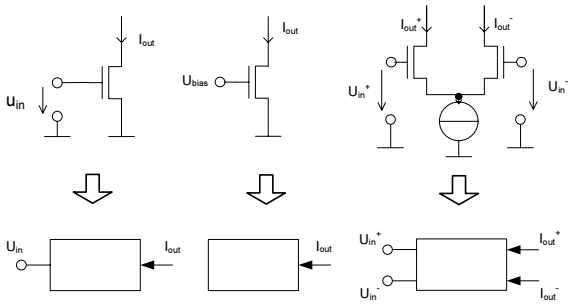

Fig. 1. Some examples of subcircuits and blocks: NMOS common-source stage, NMOS current source and NMOS differential pair

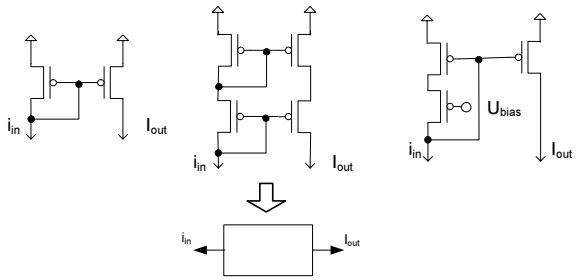

Fig. 2. Current mirrors and their block

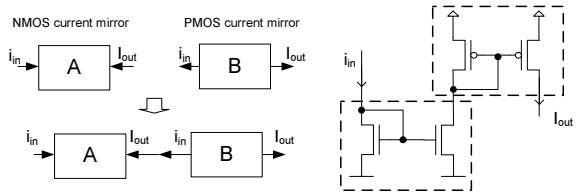

Fig. 3. Example of current rule

and then we can create plenty of block-chains (the connection of blocks in one dimension) and block-nets (the connection of blocks in two dimensions) with respect to the defined rules. These block-chains/-nets can represent the topology of circuits.

A. Features of block

The following features of terminals in each block should be defined for the later topology synthesis:

- *Type of signal*, which tells us if a voltage (U) or current (I) signal is applied to the terminal. Direction of a current signal should be given, which is determined by the bias current direction. Current signal may flow into a block or from a block.
- *Type of terminal*: which is divided into two parts - input or output terminal. The input terminal lies on the left side of the block, and the output on the right side.
- *Impedance*, which is relevant for our method. The size of impedance can be divided into two sections - low or high. For low impedance the type of signal is in lower case, and for high impedance in upper case, e.g. u means a voltage signal with low impedance, and I a current signal with high impedance.

Fig. 1 shows some examples of blocks. These blocks contain not only the transistors but also terminals of power supply (V_{dd}), bias voltage or bias current. Hence, we do not need to worry about how to connect these no signal terminals in topology synthesis. One block can be represented by different transistor level netlist (see Fig. 2). For example, basic, cascode and low power current mirror have the same type of block. After classification of basic subcircuits [8] we have got 24 types of subcircuits and 14 types of block.

B. Rules of Topology Synthesis

In this approach we have defined a set of rules conforming with the standard circuits in literature [8][9] . They describe the possible cases in which blocks can be connected. Roughly we can divide these rules into 3 categories: *general rules*, *current source rule* and *split & combination rules*.

General Rules make a reasonable connection between two blocks und create various block-chains in one dimension.

- *signal type rule*: block B can be connected to block A, if the type of input signal of block B is the same as the type of output signal of block A.
  It is obvious that a current signal from one block can flow forward into another block, only when the input signal of the second block is also current. The same is valid for voltage signals.
- *current rule*: block B can be connected to block A, if the directions of current signal of both blocks are matched and block B has low input impedance.
  As is known, an NMOS current mirror can connect to a PMOS but not to other NMOS current mirror; a PMOS diode-connected load can connect to an NMOS but not to other PMOS common-source stage. Such cases can be explained by this rule. The current of an NMOS current mirror flows into the block, and that of a PMOS in contrast from the block. At a node where an NMOS and a PMOS current mirror are connected as in Fig. 3, the directions are matched, in other words there are one current entering and one current leaving that node. Block B should also have low input impedance for proper transfer of the current signal (without bias current, i.e. small-signal).
- *Voltage rule*: block B can be connected to block A, if the output signal of block A and the input signal of block B

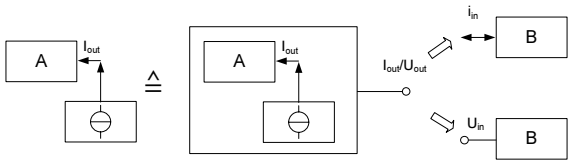

Fig. 4. Current source rule

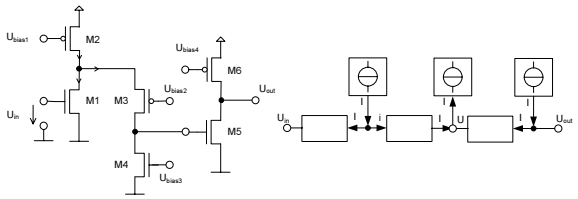

Fig. 5. Example for current source rule: folded cascode

are voltages and block B has high input impedance.
A high input impedance of a block ensures that a voltage signal will not be damped by a heavy load.

Current Source Rule can create block-chains in quasi-one dimension: The current output terminal with high impedance of a block can be connected with a current direction matched current source. Additionally, the connection node can be treated not only as a current terminal but also as a voltage terminal.

This dual feature of an output terminal is shown in Fig. 4. Treating the terminal as current terminal, we can connect a block with current input to it. The current direction of this block is irrelevant, because there are at least one current entering and one current leaving the node where blocks are connected. If it is treated as a voltage terminal, we can connect a block with voltage input to it. An example is given in Fig. 5. The structure is called folded cascode. M_1 (NMOS common-source stage) connected with M_2 (PMOS current source) is able to connect to block M_3 (NMOS cascode stage), because the current direction of M_1 and M_2 are matched. Let us see how important to have a current source in this circuit. According to the current rule in general rules a direct connection between M_1 and M_3 without M_2 is not allowed, because there are only two current leaving but none current entering the node where the blocks are connected. In order to bias M_1 and M_3, a current source must be added. Therefore this rule expands the design space. Another feature of this rule can also be explained using the example in Fig. 5. The output terminal of M_3 connected with M_4 (NMOS current source) can be treated as a voltage terminal, so that this folded cascode works as a single-stage voltage amplifier. At the right of Fig. 5 is the corresponding block-chain of folded cascode. Blocks in such chains do not really keep in line, so that we call them block-chains in quasi-one dimension.

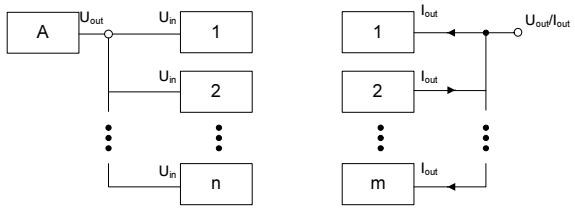

Fig. 6. Split & combination rules

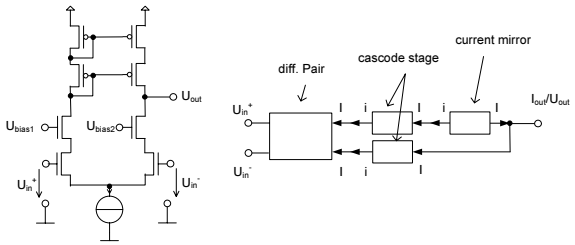

Fig. 7. cascode op amp and its block-net

Split & Combination Rules expand the connection between blocks in two dimensions and we get a plenty of block-nets, which represent analog circuits.

- *Voltage split rule*: a voltage output signal of a block can be split to feed several blocks.
- *Current combination rule*: output current signals of blocks can be combined to an input of a block. If there is no bias current entering or leaving that node where all blocks are connected, one current source should be added.

These rules are described in Fig. 6. In order to control the complexity of synthesized circuits a reasonable number of blocks, which the voltage signal spilt to or whose current signals can be combined, is smaller then 3.

A common application area of these rules is to determine, how to connect the blocks behind the differential pair. As we see in Fig. 1, there are two current outputs for differential pair block. They can connect to other blocks separately. It seems like that differential pair block is followed by two block-chains. If both block-chains have current outputs with high impedance, both block-chains can be connected together. If both are voltage outputs, they build a differential signal, which can be connected to another differential pair. An example for the first stage of a cascode operational amplifier (schematic and its block-net), also called unbuffered op amp or OTA [9] , is given in Fig. 7.

For a further synthesis with differential pairs there are some other rules to define: one of the differential pair outputs with a

bias current flowing out of the block can directly be connected to GND (ground), and one with a bias current flowing into block can be connected to V_{dd} (power supply); one of differential pair inputs can directly be connected to GND.

C. Topology Generation Step

In this section, we will describe how to create circuit topologies by using predefined hierarchical blocks and specification.

- *Block specification*: If we treat block-chain or -net as "black box", then all what we know about this black box are only input and output terminals but not its structure or function. We need the following input information for the op amp in Fig. 7: differential voltage signal at input terminals with high impedance and voltage signal at output terminal with high impedance. The structure of block-chain/-net can not be influenced by user, because they are generated according to synthesis rules. Therefore the input for the topology generator are just the attributes of terminals. This input information can be applied to create a large variety of topologies, but it is also a challenge to select an appropriate circuit from them. This task will be done in the topology selection phase using symbolic analysis. Additional input information is the maximum number of blocks in block-chain or -net, so that the size of a topology and the number of synthesized topologies can be limited. The block size of the op amp shown in Fig. 7 is for example 4.
- *Algorithm*: A structure chart of chaining up blocks is shown in Fig. 8. The algorithm writes one or several corresponding circuits in VHDL-AMS or Spice netlist from a block-chain/-net, since a block can represent one or more analog subcircuits. If we have got a block-net for Fig. 7 from the generator, another op amp with basic current mirror can be also a variant.

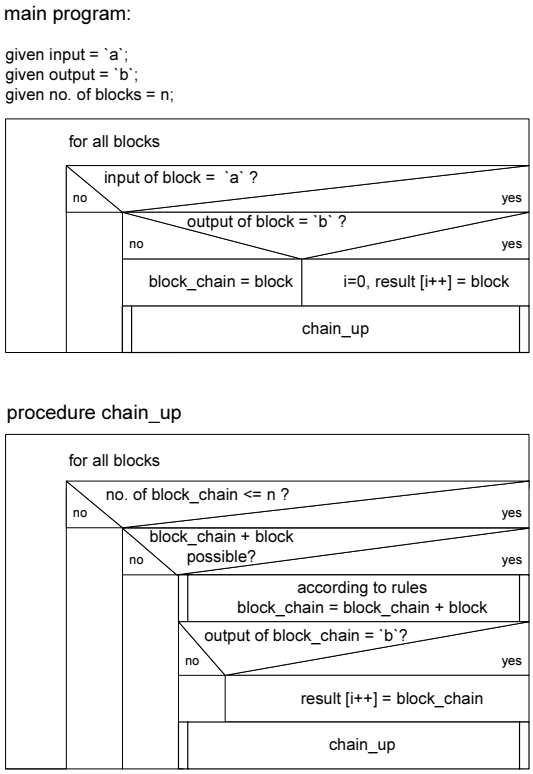

Fig. 8. structure chart of Algorithm

III. Performance Estimation by Symbolic Analysis

The topology generation steps generate all possible topologies which meet the rules defined in previous section. The rules restrict the possible structure space implicitly or explicitly to more or less usable circuits. However, after this step a very high number of circuit are good candidates for the design goal. This number has to be reduced to the real feasible circuits. Common topology synthesis approaches use Spice simulations in the loop [5]. Furthermore a full, time consuming sizing of circuit parameters is needed to be able to evaluate the performance of each circuit.

We propose a stepwise method based on symbolic analysis to come up with these time consuming task and split it into several hierarchical steps to detect at a very early stage non sufficient circuit topologies:

a) Calculate symbolic formulas for linear performances in terms of linear device parameters.

b) Compute for each individual performance formulas in terms of the independent sizing parameters width, length and bias current of (pairs of) transistors.

c) Check the performances against the specification.

d) Do detailed sizing including biasing for the remaining circuits. This step is not included in this paper and will be future work.

Step a), b) and c) can be executed together for each performance to quickly get false results and prevent the time consuming calculation of other performances. In the following, each step is explained for the example of an operational amplifier.

A. Specification

Input for the analysis are n_{Spec} specifications in terms of performances using inequalities:

$$S = \begin{bmatrix} Gain & > & 60\,dB \\ GBW & > & 1 \cdot 10^9\,Hz \\ PSRR & > & 40\,dB \\ R_{out} & > & 100\,k\Omega \end{bmatrix} \quad (1)$$

On the other hand we have basic device and biasing parameters in predefined ranges according to process technology, which are used for sizing.

$$\begin{array}{rcl} P & = & [W_1, L_1, I_{Bias1}, \ldots, W_n, L_n, I_{Biasn}, \ldots] \\ \text{with} & \forall i & W_i \in [W_{min}..W_{max}],\ L_i \in [L_{min}..L_{max}], \\ & & I_{Biasi} \in [I_{Biasmin}..I_{Biasmax}] \end{array} \tag{2}$$

where W_i, L_i, I_{Biasi} are the width, length and bias current of the transistors of the i-th block and n is the block count. Hence, matching is automatically ensured for each block or transistor pair.

B. Linear Symbolic Analysis

Linear symbolic analysis is a technique for calculating transfer functions in frequency domain in symbolic form [10] [11]. The basis for the analysis is a netlist with transistors being replaced by their linear model using small signal parameters like transconductance g_{mi}, output resistance r_{DSi} and capacitances. Additionally, a test bench is provided to correctly model the environment of the circuit. Using a modified nodal approach a system of differential algebraic equations is setup. We use a technique described in [12] to simplify this system and finally solve it for the output variables. The simplification process needs assumptions about numerical values of the small signal parameters which can easily estimated from technology parameters. We use center values of ranges in equation (2) calculate the correponding small signal values using a nonlinear device model. The result is a transfer function in terms of small signal parameters:

$$\begin{array}{l} H_l(s) = \frac{Y(s)}{X(s)} = \\ f(g_{m1}, \ldots g_{mn}, r_{DS1}, \ldots r_{DSn}, c_{GS1}, \ldots c_{GSn}, \ldots) \end{array} \tag{3}$$

where $Y(s)$ is the output and $X(s)$ the input of a performance function, for example U_{output} and I_{output} respectively for calculating the output resistance. A symbolic formula for the corner frequency f_c could be derived in a similar way.

C. Performance evaluation

In order to correctly check the performances with respect to matching, same bias currents and dependence between g_m, r_{DS}, W, L the linear transfer function is modified by inserting basic formulas for the small signal parameters in terms of basic device parameters derived from a nonlinear device model.

$$\begin{array}{l} H_d(s) = \frac{Y(s)}{X(s)} = \\ f(W_1, \ldots W_n, L_1, \ldots L_n, I_{bias1}, \ldots I_{Biasn}, \ldots) \end{array} \tag{4}$$

Finally the transfer functions is checked in ranges for fulfilling the specification. That means evaluating for each specification the performance equation (4) for all corners of a lower dimensional subset of the parameter space P and checking if a feasible solution exists.

These method is quick because in general we have restricted number of parameters in the transfer function, which reduce the dimension of the parameter space. In contrast, a full Spice-based performance evaluation needs to check all corners of all parameters resulting in a basic sizing step (at least the bias voltages should be sized), DC-analysis and AC-analysis step for each corner, which seems to be prohibitive in runtime. In our case the runtimes for symbolic analysis is much larger than the following corner analysis. A small example of the symbolic analysis will be shown in Section IV.

IV. Synthesis Results

The proposed method was applied to generate and select different circuits types, such as unbuffered op amp and differential-input current amplifier. Using PC with 3GHz prozessor we need $1-2$ minutes to create all circuits and about 2 seconds for analysis one circiuts.

A. Unbuffered op amp

A general application area of analog synthesis is the design of operational amplifiers. From these essential elementary circuits several useful circuits, such as comparator, differential ring oscillator and filters can be derived. The op amps with a differential input and a high output impedance are classified as unbuffered op amps. They are always used as the first stages of op amps. In this section, we synthesized thousand of op amps and selected several appropriate op amps.

The input information for hierarchical method is listed as follows:

- *Input terminals*: A differential voltage signal with high impedance
- *Output terminal*: A voltage signal with high impedance
- *No. of blocks*: 4
- *Rules*: Without voltage split rule, in order to reduce the complexity of the circuits

The following values of technology parameters were chosen for symbolic analysis: $W, L = [1\,10^{-7}..2\,10^{-5}]$, $I_{Bias} = [0.01\,10^{-6}..0.2\,10^{-6}]$, $K_p = 400\,10^{-6}$, $V_E = 5\,10^5$, $c_{gd} = 1pF$, $c_{gs} = 1pF$. The electrical specification for topology selection is described as follows: *DC gain* is more than $80\,dB$, *Corner frequency* is more then $1\,MHz$ and *Output impedance* is more then $400\,k\Omega$.

In topology generation step we have got total 448 blockchains/-nets, which can represent 2732 unsized circuits, because some blocks respond to more then one subcircuits (e.g. basic and cascode current mirror). In the next selection step all these circuits were analyzed and evaluated. Finally 50 circuits were chosen. Some of them have familiar topologies because of complimentary characteristics, i.e. PMOS subcircuits instead of NMOS subcircuits or vice versa. The reason why we didn't choose only one circuit but several circuits is, that the designer can get various supply of circuit topologies for the later sizing step. Here one circuit of them is shown in Fig. 9. And its performance was calculated: DC gain $81.3\,dB$, corner frequency $1.01\,MHz$, output impedance $5\,M\Omega$.

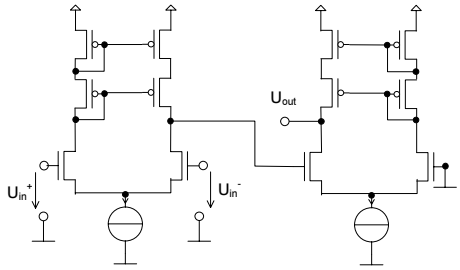

Fig. 9. Op amp generated by synthesis method

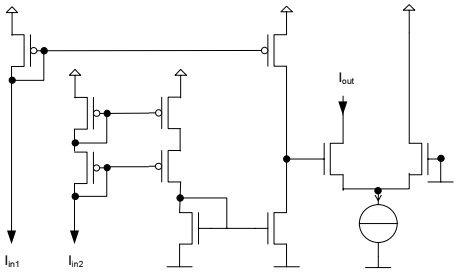

Fig. 10. Differential-input current amplifier generated by synthesis method

B. Differential-input current amplifier

A current amplifier is an amplifier with a defined relationship between the input and output currents, which offers wide signal dynamic range and will find applications in low-voltage and in switched-current circuits. We used the following input information for topology generation:

- *Input terminals*: A differential current signal with low impedance and the bias current flows from the circuits
- *Output terminal*: A current signal with high impedance and the bias current flows into the circuits
- *No. of blocks*: 4
- *Rules*: Without voltage split rule, in order to reduce the complexity of the circuits

The following values of small-signal CMOS model parameters were chosen for symbolic analysis: $g_m = 0.0021S$, $r_{ds} = 100k\Omega$, $c_{gd} = 1pF$, $c_{gs} = 1pF$. In this example we do not use range values for device parameters resulting in only nominal point analysis.

The electrical specification for topology selection is described as follows: *DC gain* should be as high as possible, *Corner frequency* is more then $2MHz$, *Input impedance* is less then $1k\Omega$ and *Output impedance* is more then $100k\Omega$.

There were total 1157 generated block-chains/nets according to 4718 circuits. In Fig. 10 we show one example circuit of 37 selected circuits.

V. Conclusion

This paper has introduced a new concept of analog circuit synthesis using hierarchical blocks for topology generation and using symbolic analysis for topology selection. This hierarchical method enables the designer to completely search the topology space with restrictions given by rules. The rules extracts the essential standard from human circuit design. Symbolic analysis helps designer choose some appropriate circuits from a large number of generated topologies. A further step towards an completely automated method would be a subsequent automated sizing process, which will be part of our future work. We have successfully demonstrated two applications, op amp and current amplifier.

References

[1] Georges G. E. Gielen and Rob A. Rutenbar, "Computer-Aided design of analog and mixed-Signal integrated circuits" *Proceedings of the IEEE*, Vol. 88, No. 12, Dec. 2000, pp. 1825-1852

[2] R. Harjani, R. Rutenbar and L. R. Carley, "OASYS: A framework for analog circuit synthesis" *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 1247-1265, Dec.1989

[3] F. El-Turky and E. Perry, "BLADES: An artificial intelligence approach to analog circuit design" *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 680-691, June 1989

[4] W. Kruiskamp and D. Leenaerts, "DARWIN: CMOS opamp synthesis by means of a genetic algorthm" in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, 1995, pp. 550-553

[5] T. R. Dastidar, P. P. Chakrabarti and R. Ray, "A synthesis system for analog circuits based on evolutionary search and topological reuse" *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, Apr. 2005, pp. 211-224

[6] E.A.M. Klumperink, F. Bruccoleri, B. Nauta, "Finding all Elementary Circuits Exploiting Transconductance", *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 1, 6-9 May 2001 pp. 667 - 670

[7] H. Graeb, S. Zizala, J. Eckmueller and K. Antreich, "The sizing rules method for analog integrated circuit design" *ICCAD*, 2001

[8] Behzad Razavi, *Design of Analog CMOS Integrated Circuits* [Boston, Mass.]: McGraw-Hill, 2000

[9] P.E. Allen and D.R. Holberg, *CMOS Analog Circuit Design* London, U.K.: Oxford Univ. Press, 2002

[10] F.V. Fernandez, A. Rodriguez-Vazquez, J.L. Huertas and G. Gielen: "Symbolic Analysis Techniques - Applications to Analog Design", Automation. *IEEE Press*, 1998

[11] G. Gielen and W. Sansen: "Symbolic Analysis for Automated Design of Analog Integrated Circuits" *Kluwer Academic Publishers, Boston/ Dordrecht/London* , 1991

[12] Popp, R. and Barke, E.: "Symbolic Analysis of Nonlinear Analog Circuits by Simplification of Nested Expressions" *SMACD 00: Proc. of the InternationalWorkshop on Symbolic Methods and Applications in Circuit Design*, 2000