Parameter Controlled Automatic Symbolic Analysis of Nonlinear Analog Circuits

Ralf Popp, Joerg Oehmen, Lars Hedrich, Erich Barke
Institute of Microelectronic Circuits and Systems, University of Hannover, Germany
[popp,oehmen,hedrich,barke]@ims.uni-hannover.de

Abstract

In this paper we introduce an approach for parameter controlled symbolic analysis of nonlinear analog circuits. Based on a state-of–the-art algorithm, it enables the removal of specific circuit parameters from a symbolic circuit description, given as a set of nonlinear differential algebraic equations (DAEs). During the removal, singularities are considered, which includes structural changes of the set of DAEs. The feasibility of our approach is shown by several circuit examples.

1. Introduction

Symbolic analysis is a powerful method to describe specifications or behavior of analog circuits as a function where its physical parameters and variables are represented by symbols. Since the early years of its development it was restricted to linear circuits [1]. Later, symbolic analysis was extended first to weakly nonlinear [2] and then to strongly nonlinear circuits [3]. While symbolic analysis of linear circuits received growing interest among circuit designers, the nonlinear part, due to its limits, remains a topic of research [4], [5], [6].

In this paper, we present a new approach, which allows symbolic analysis of nonlinear circuits controlled not only by a given error bound but also by the circuits' parameters. This means, that the remaining parameters in a simplified function will no longer be chosen arbitrarily: Parameters, which are removable due to the desired accuracy, are guaranteed to be removed.

Within our approach, which is based on the method in [3], we introduce a new kind of symbolic simplification operation. This operation tries to remove certain or all parameters from a symbolic circuit. It is able to handle singularities resulting from a parameter's removal, even if this requires structural changes in the circuit description.

In the following second section, the underlying method is described briefly. In Section 3, the new simplification algorithm is explained. The feasibility of the approach is shown, using several examples, in Section 4.

2. Method

We use an automatic simplification procedure to apply symbolic analysis to nonlinear analog circuits. The concept behind our approach is shown in Figure 1:

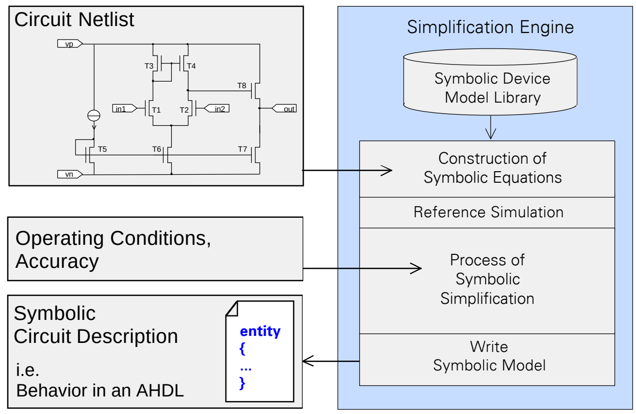

Fig. 1: Symbolic simplification concept

The procedure uses a simplification engine, which works fully automatically on a SPICE netlist and some user defined parameters. From the SPICE netlist a set of symbolic differential algebraic equations (set of DAEs) is set up, using the modified nodal approach. For this purpose, a library of appropriate symbolic model equations for all circuit elements has been developed. The created set of DAEs is reduced by a simplification process, which will be explained in detail in this section. As a result a symbolic circuit description is obtained, which consists of a new set of equations with a drastically reduced number of variables and terms.

The simplification process itself consists of various types of symbolic simplification operations, performed sequentially in a loop, shown in Figure 2. Each type of simplification operation can be divided into three parts:

- Preprocessing of a virtual symbolic simplification
- Virtual symbolic simplification and verification.
- Real symbolic simplification in a postprocessing

In the first part, a number of symbolic bits a_k, $a_k \in \{0,1\}$ is inserted into the set of DAEs. The rules for the insertion correspond to the performed type of simplification and follow one main principle: Assigning zero to such a symbolic bit causes a symbolic simplification, while setting it to one leaves the set of DAEs untouched. For example, in (1) the inserted symbolic bits a_1 and a_2 correspond to the removal of the first and the second addend of that equation:

$$a_1 \cdot u_1(t) + a_2 \cdot u_1(t) \cdot u_2(t) + \ldots = 0 \qquad (1)$$

This "modeling" of symbolic simplifications does not imply any restrictions concerning the variety of possible types of simplification. All simplification operations remain possible. Examples are shown in [6] and in this paper. Further, the insertion of symbolic bits offers the opportunity to transform the complex process of symbolic simplification into a simple value assignment process. Therefore, simulators are able to perform several different virtual symbolic simplifications by changing numerical values of the symbolic bits.

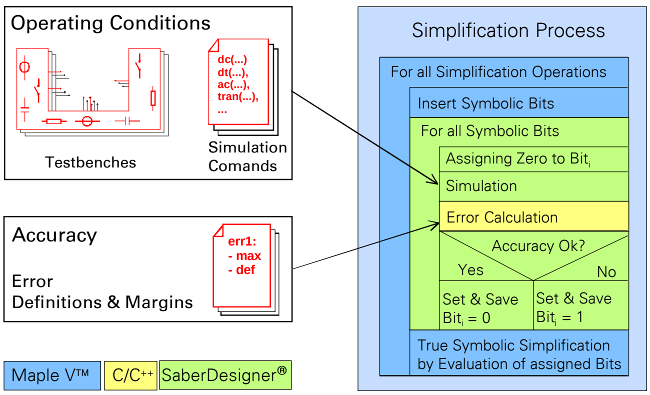

Fig. 2: The simplification process

In the second part, performed by a numerical simulator, these symbolic bits are set to zero, one by one within a loop. Corresponding to each symbolic bit set to zero, the desired accuracy of the circuit description is checked with respect to the input/output behavior under the user-defined operating conditions of the circuit. The operating conditions are defined by one or several testbenches, which are connected to the circuit, and by the corresponding simulations to be performed. As a result of the loop a vector $\vec{a}$, $\in \Omega^m$, $\Omega = \{0,1\}$ is obtained, indicating those symbolic bits, which can be assigned zero without exceeding the user defined accuracy.

In the third part the actual symbolic simplifications, which do not exceed the user defined accuracy, are performed, due to the information in $\vec{a}$. Additionally, a symbolic postprocessing is executed, which eliminates variables and equations respectively by substitution.

3. Parameter Controlled Simplifications

For symbolic analysis, complex transistor models are used containing up to 200 parameters and 50 equations to model various physical effects. They are formulated as a 'single expression', meaning that all MOS model equations are simultaneously valid in all operating regions.

Often a transistor is operating in a certain region, in which some physical effects do not occur. According to that, some device parameters may be useless in a particular circuit description and can be removed without causing a significant error. The new parameter controlled simplification is able to remove specific parameters from the set of DAEs, not dependent of arbitrarily performed hierarchic simplifications. This is possible, even if structural changes of the set of DAEs are required.

3.1. Algorithm

The parameter oriented simplification operation simplifies expressions by removing parameters from the circuit description. It considers two cases, which lead in fact to two simplification operations, the removal of a parameter by setting it to zero and by setting it to infinity. As every simplification operation, both are divided into the three parts mentioned above.

Figure 3 illustrates the preprocessing algorithm for a virtual symbolic simplification by insertion of symbolic bits a_i:

for each value *v* in $\{0,\propto\}$in **do**
{
  for each parameter p_i **do**
  {
    insert a_i by a substitution: $p_i = f(p_i, a_i, v)$
    if $a_i = 0$ causes a singular addend in an equation **then**
    {
      if *limit of addend(f(*p_i *® v)* exists and it is finite **then**
        modify $f(p_i,a_i)$;
      else if *addend* represents a current through p_i **then**
        transform *set of DAEs*;
      else
        take back substitution of p_i by $f(p_i,a_i)$
    }
  }
}

Fig. 3: Algorithm of the insertion of symbolic bits

For each parameter p_i, the insertion of a symbolic bit a_i is performed by a substitution ($p_i = f(p_i, a_i, v)$). Then, it is checked if the assignment of zero to a_i causes a singularity in any toplevel addend in the set of DAEs which contains f. Three possibilities exist to avoid such a singularity: First it is possible to perform a modification of f, if

$$\lim_{p_i \to v}\left(addend\right) \tag{2}$$

exists and is finite. Second, the singularity can be prevented by a transformation of the set of DAEs, if p_i represents a circuit element and if the singular addend represents a current through p_i. And third, the singularity can be avoided by taking back the substitution of p_i, which means to cancel its removal.

3.1.1. Insertion of symbolic bits

According to the two cases of the parameter controlled simplification operation, two different modes of insertion of symbolic bits are possible. This is illustrated in the following expression taken from the BSIM3 transistor model:

$$\frac{D_{VT1w}\cdot W_{eff}(t)\cdot L_{eff}(t)}{2\cdot\sqrt{\sqrt{\frac{2\cdot e_{Si}\left(\Phi_S - V_{bseff}(t)\right)}{q\cdot N_{ch}}}}\cdot T_{ox}\cdot\left(1+D_{VT2w}\cdot V_{bseff}(t)\right)} \tag{3}$$

It consists of four variables and seven parameters. The two modes of insertion of symbolic bits lead to two different expressions, in (4) the expression considering the parameters' removal by setting them to zero:

$$\frac{a_1 D_{VT1w} \cdot W_{eff}(t) \cdot L_{eff}(t)}{2 \cdot \sqrt{\sqrt{\frac{2 \cdot a_2 e_{Si}\left(a_3 \Phi_S - V_{bseff}(t)\right)}{a_4 q \cdot a_5 N_{ch}}} \cdot a_6 T_{ox} \cdot \left(1 + a_7 D_{VT2w} \cdot V_{bseff}(t)\right)}} \quad (4)$$

and in (5) the second expression considering the parameters' removal by setting them to infinity:

$$\frac{\frac{D_{VT1w}}{a_1} \cdot W_{eff}(t) \cdot L_{eff}(t)}{2 \cdot \sqrt{\sqrt{\frac{2 \cdot \frac{e_{Si}}{a_2}\left(\frac{\Phi_S}{a_3} - V_{bseff}(t)\right)}{\frac{q}{a_4} \cdot \frac{N_{ch}}{a_5}}} \cdot \frac{T_{ox}}{a_6} \cdot \left(1 + \frac{D_{VT2w}}{a_7} \cdot V_{bseff}(t)\right)}} \quad (5)$$

Thus, the symbolic removal of the i-th parameter by setting it to zero or to infinity can be performed virtually, by setting the a_i to zero. In this example singularities arise by assigning zero to a_3 and a_4 in (4) or by assigning zero to any of the a_i in (5).

3.1.2. Modification of the insertion of symbolic bits

In order to avoid singularities their type has to be checked. To distinguish different types of singularities, two conditions are examined as shown in Figure 3. First, the limit in (2) is checked whether it exists and is finite. As the insertion of symbolic bits is implemented in Maple V™, a built-in command for calculating the limit [7] has been used. If the limit exists and it is finite, the singularity can be avoided by a modification of the insertion of the corresponding symbolic bit:

$$addend(a_1) \Rightarrow a_1 \cdot addend + (1 - a_1) \cdot \lim_{p_i \to n}(addend) \quad (6)$$

By assigning a_1 to one, the original addend remains unchanged and by assigning a_1 to zero it is replaced by the limit of the singular addend.

3.1.3. Transformation of the set of DAEs

If the limit does not exist, it is checked if p_i represents a circuit element, like R represents a resistor and if the toplevel addends containing p_i represent a current through that element. In that case, the removal of this parameter causes a short inside the circuit description. Therefore, the set of DAEs has to be modified in order to model the short. This is performed by connecting the two nodal voltages, which represent the two adjacent nodes of the short. Two nodal equations of the set of DAEs have to be combined eliminating one equation and one variable. Due to the fact, that this simplification is embedded into the whole simplification process, it has to be realized by a insertion of symbolic bits too.

This request leads to a complex procedure, because it is not known during the insertion of symbolic bits, which element will be removable in the end. Every possible removal of an element has to be considered. For example if there are two resistors in series: All three nodal voltages, connected to a resistor have to be prepared for connection with each other by inserting symbolic bits. This leads to complex expressions if there are several resistors connected to each other or in a circle. To illustrate this, the procedure of preparing the connection of nodal voltages will be explained by an example, shown in Figure 4.

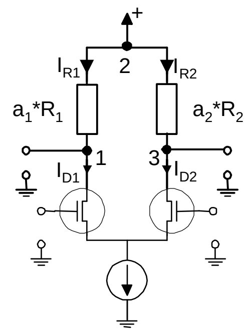

Fig. 4: Schematic of a differential pair

This differential pair, contains two resistors R_1 and R_2 to be removed, which is indicated by the symbolic bits a_1 and a_2. Kirchhoff's nodal equations of the circuit are shown in (7). To get concise expressions the sum of the currents flowing into the i-th node is abbreviated by S_i :

$$\begin{aligned} S_1 &= I_{out1} + I_{R1} - I_{D1} = 0 \\ S_2 &= I_{VCC} - I_{R1} - I_{R2} = 0 \\ S_3 &= I_{out2} + I_{R2} - I_{D2} = 0 \end{aligned} \quad \text{with} \quad \begin{aligned} I_{R1} &= \frac{u_{n2}(t) - u_{n1}(t)}{a_1 \cdot R_1} \\ I_{R2} &= \frac{u_{n2}(t) - u_{n3}(t)}{a_2 \cdot R_2} \end{aligned} \quad (7)$$

As expected, I_{R1} and I_{R2} become singular if a_1 or a_2 respectively, is set to zero.

To prepare the removal of resistor R_1 and the connection of the nodes 1 and 2 respectively, the nodal equation equations have to be transformed as:

$$\begin{aligned} S_1 + (1 - a_1) \cdot S_2 &= 0 \\ (1 - a_1) \cdot \left(u_{n2}(t) - (1 - a_1) \cdot u_{n1}(t) \right) + a_1 \cdot S_2 &= 0 \\ S_3 &= 0 \end{aligned} \quad (8)$$

This kind of insertion of a_1 either allows to leave the equations untouched by assigning a_1 to one or to combine the first two equations by assigning a_1 to zero. In that case, the first equation results in the sum of all currents flowing into both nodes 1 and 2, leading to two canceling terms I_{R1}. The second equation becomes a potential equation, which is removable later by substitution during the symbolic postprocessing. If both resistors R_1 and R_2 have to be prepared for elimination the set of equations becomes more complex, as shown in (9):

$$\begin{aligned} S_1 + \overline{a_1} \cdot \left(S_2 + \overline{a_2} \cdot S_3\right) &= 0 \\ \overline{a_1} \cdot \left(u_{n2}(t) - \overline{a_1} \cdot u_{n1}(t) \right) + a_1 \cdot \left(S_2 + \left(a_1 \wedge \overline{a_2}\right) \cdot S_3\right) &= 0 \\ \overline{\left(\left(a_1 \wedge a_2\right) \vee a_2\right)} \cdot & \\ \left(u_{n3}(t) - \left(\overline{a_1} \wedge \overline{a_2}\right) \cdot u_{n2}(t) - \overline{\left(\overline{a_1} \wedge \overline{a_2}\right)} \cdot \overline{a_2} \cdot u_{n2}(t) \right) & \\ + \left(\left(a_1 \wedge a_2\right) \vee a_2\right) \cdot S_3 &= 0 \end{aligned} \quad (9)$$

where boolean functions like

$$\begin{aligned} \overline{a_1} &= (1 - a_1) \\ a_1 \wedge a_2 &= a_1 \cdot a_2 \\ a_1 \vee a_2 &= sign(a_1 + a_2) \end{aligned} \tag{10}$$

have been used, to keep the equations short. This example shows the preparation of resistors' removal, preventing singularities in the set of DAEs. All singularities which are evoked by the removal of parameters representing circuit elements like resistors and capacitances are avoided in this way.

3.1.4. Symbolic postprocessing

The third part of the simplification operation is the symbolic postprocessing. Further reductions, like the substitution of variables, are performed here. For example, if the two resistors in Figure 4 are removable, (9) becomes

$$\begin{aligned} S_1 + S_2 + S_3 &= 0 \\ u_{n2}(t) - u_{n1}(t) &= 0 \\ u_{n3}(t) - u_{n2}(t) &= 0 \end{aligned} \tag{11}$$

which means that a_1 and a_2 can be assigned to zero. Due to the canceling terms for I_{R1} and I_{R2} within the sum of currents, the parameters R_1 and R_2 already have been removed from the set of equations. Apart from that, neither a variable nor an equation have been eliminated yet. Therefore, within the postprocessing, the second and third equation of (11) are eliminated and the variable $u_{n3}(t)$ is substituted by $u_{n2}(t)$ and $u_{n2}(t)$ is substituted by $u_{n1}(t)$ in the whole set of DAEs. In this way, all equations, which look like

$$u_{ni}(t) - u_{nj}(t) = 0 \tag{12}$$

are eliminated by substitution of variables. This includes not only those potential equations which arise from removing resistors or capacitors. Especially equations describing an inductor can be simplified in a similar way, as the following example shown in Figure 5 will show:

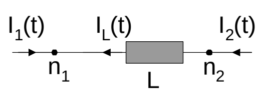

Fig.5: Inductor

After the insertion of the symbolic bits we obtain a set of three equations:

$$\begin{aligned} I_L(t) + I_1(t) &= 0 \\ I_2(t) - I_L(t) &= 0 \\ u_{n1}(t) - u_{n2}(t) + \frac{d}{dt}\left(a_1 \cdot L \cdot I_L(t)\right) &= 0 \end{aligned} \tag{13}$$

It contains two nodal equations for each node (n_1 and n_2) of the inductor and an additional equation for the additional variable $I_L(t)$, which represents the current through the inductor and which results from the MNA used to build the set of DAEs in the beginning.

Performing a symbolic simplification corresponding the setting of a_1 to zero in (13) also results in an equation like (12). But in this case, more reductions are possible than just the substitution of $u_{n1}(t)$ by $u_{n2}(t)$. This results from the fact, that there are two nodal equations containing the current $I_L(t)$, which can be eliminated by adding one equation to the other – eliminating variable $I_L(t)$ and one equation.

Finally, (13) results in:

$$I_1(t) + I_2(t) = 0 \tag{14}$$

Using the combination of the described procedures, specific symbolic parameters can be removed from the set of DAEs, as to be shown in the next section.

4. Results

Two different circuits have been analyzed using the simplification procedure described in Figure 1. Their schematics and operating conditions, consisting of the testbench and the performed simulations are shown in Figure 6 and Figure 7, respectively. The circuits are printed in black, their invoking testbenches are printed in gray.

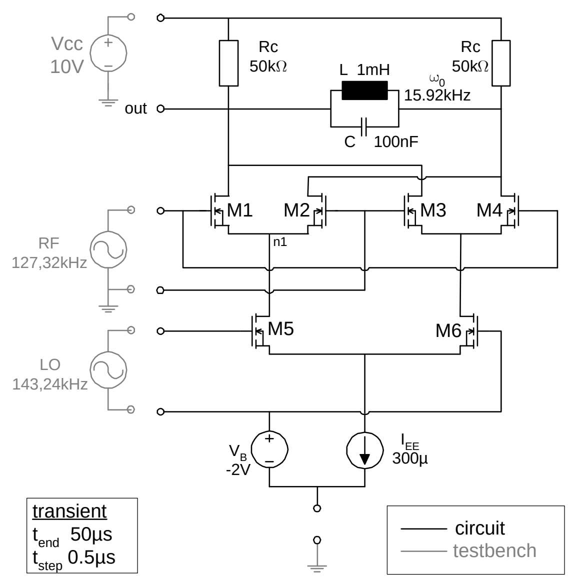

Fig.6: MOS Gilbert cell and its testbench

Figure 6 shows a MOS Gilbert multiplier cell [8] with an output resonance circuit in MOS technology. The transistors are modeled by a simplified BSIM3 model. The Gilbert cell is embedded in a testbench consisting of three voltage sources, performing a transient simulation over 50 µs. As a second circuit, Figure 7 shows a differential pair in bipolar technology, with transistors modeled by the Gummel-Poon model. It is embedded in a testbench consisting of three voltage sources, performing a DC-transfer simulation and multiple AC simulations at three operating points.

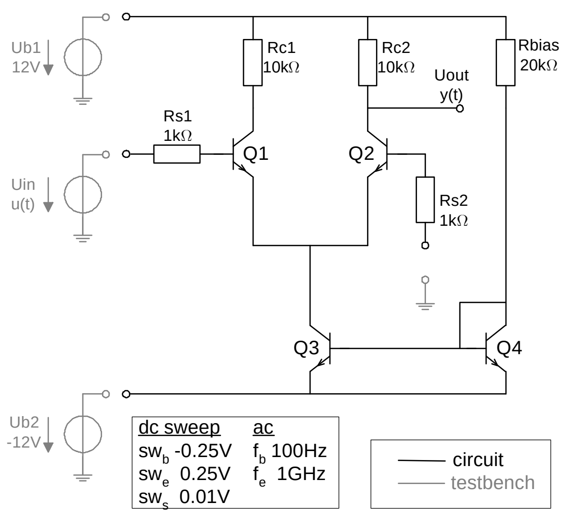

Fig.7: Differential pair and its testbench

To show the benefits of the parameter controlled approach, the symbolic analysis has been performed with and without the new simplification operation. Two different error margins have been used to define a desired accuracy. Table 1 shows the results, a comparison between the original and two differently simplified sets of DAEs, all describing the behavior of the appropriate circuit.

Gilbert cell Property	Original	Simplified			
		without "Param"		with "Param"	
Error Margin		2.5%	5%	2.5%	5%
# Equations	72	54	48	53	46
# Parameter	78	56	40	44	36
TRC	319002	95900	61730	74658	57116

Differential pair Property	Original	Simplified			
		without "Param"		with "Param"	
Error Margin		2.5%	5%	2.5%	5%
# Equations	34	28	27	19	18
# Parameter	65	25	24	16	15
TRC	553816	55825	52523	18608	17108

Table 1: Results of simplification

Measured are the number of equations and the number of parameters in the sets of DAEs and a total runtime coefficient (TRC). The TRC is an empirical measure, which calculates the complexity of a set of DAEs based on the weighted number of operations and variables:

$$TRC = \begin{pmatrix} n_{\exp}\cdot 10 + n_{power}\cdot 10 \\ + n_{derivatives}\cdot 6 + n_{multiplications}\cdot 5 \\ + n_{heavysides}\cdot 5 + n_{additions}\cdot 1 \\ + n_{other\ functions}\cdot 15 \end{pmatrix} \cdot n_{independant\ vars}^{1.5} \qquad (15)$$

where n_x is the number of the operation x. All measurements show a significant increase of reduction ratio due to the parameter controlled simplification operation. Concerning the TRC of the differential pair the reduction ratio due to the parameter oriented simplification even increases from about 10 to about 30 in both cases of accuracy.

Of course, this reduction ratio neither leads to interpretable formulas for design assistance, nor generates fast behavioral models, yet. But the parameter controlled method is a good step to attain both goals finally.

Observing the in fact increasing number of removed parameters, leads to two remarkable but expected results: In the bipolar circuit all base and collector resistances of the transistors have been removed by the new algorithm. Due to the feedback loop, the emitter resistors were not removable. In the MOS circuit, the additionally removed parameters were capacitances, which have been contained in denominators of the circuit description. Both types of parameters have not been removable before, because of the singularities, which arose from their removal.

5. Conclusion

A parameter controlled simplification algorithm has been presented, which has been embedded into a state-of-the-art method for symbolic analysis of nonlinear analog circuits. The new algorithm opens the opportunity not only to eliminate specific parameters, but also to modify the circuit description in order to avoid singularities, which arise from a parameters removal. Thus it is able to eliminate any circuit parameter, if the desired accuracy is not violated. The feasibility of the approach has been illustrated by two circuit examples in different technologies.

References

[1] F.V. Fernandez, et. al., *"Symbolic Analysis Techniques -- Applications to Analog Design"*, IEEE Press, 1998

[2] P. Wambacq, J. Vanthienen, G. Gielen, W. Sansen, *"A Design Tool for Weakly Nonlinear Analog Integrated Circuits with Multiple Inputs"*, IEEE Custom Integrated Circuits Conference, pp. 5.1.1-5.1.4, 1991

[3] C. Borchers, L. Hedrich, E. Barke, *"Equation-Based Behavioral Model Generation for Nonlinear Analog Circuits"*, *Proc. DAC*, pp. 236-239, 1996

[4] R. Sommer, E. Hennig, T. Halfmann, T. Wichmann, *"Symbolic Modeling and Analysis of Analog Integrated Circuits"*, Proc. ECCTD'99, pp. 66-69, Stresa, Italy, 1999

[5] M. Olbrich, R. Popp, l. Näthke, L. Hedrich, E. Barke *"A Combined Structural and Symbolic Method for Behavioral Modeling of Nonlinear Analog Circuits"*, Proc. ECCTD'01, pp. 274-278, Espoo, Finland, 2001

[6] R. Popp, E. Barke: *"Symbolic Analysis of Nonlinear Analog Circuits by Simplification of Nested Expressions"*, Proc. Int. Workshop on Symbolic Methods and Applications in Circuit Design, SMACD'00, pp. 151-154, Lissabon, 2000

[7] K.O. Geddes, *"A New Algorithm for Computing Symbolic Limits Using Generalized Hierarchical Series"*, Proceedings of ISSAC, 1998

[8] E. W. Greeneich, *"Analog Integrated Circuits"*, Chapman & Hall, 1997