

# Symbolic Analysis of Nonlinear Analog Circuits by Simplification of Nested Expressions

Ralf Popp Erich Barke

Institute of Microelectronic Systems  
University of Hanover, Appelstr. 4, D-30167 Hanover, Germany  
E-Mail: popp@ims.uni-hannover.de

## ABSTRACT

In this paper we present an approach for symbolic analysis of strongly nonlinear analog circuits including MOS devices. It is based on a method, that applies symbolic simplification to generate nonlinear behavioral models for bipolar circuits. Our approach allows extended simplifications, well suited for the nested structure of the large addends in MOS circuit equations.

## INTRODUCTION

Although the interest in symbolic analysis of analog circuits has grown, it is still mostly restricted to linear [1], [2], [3] and weakly nonlinear [4] applications. Symbolic analysis of circuits with strong nonlinearities has been restricted first to circuits with bipolar devices [5], [6], [7].

In this paper we present an approach for symbolic simplification of strongly nonlinear analog circuits including MOS devices. The approach is based on a method, mentioned above [5], that applies symbolic simplification to generate nonlinear behavioral models for bipolar circuits.

Our approach concerns a new kind of symbolic simplification. It has been developed with regard to MOS device model equations, like BSIM3v3. These equations typically consist of large addends, which can not be simplified as a whole. Therefore, the nested parts of those large terms have to be simplified separately. Our contribution will show how this can be done and which simplification results have been achieved.

## METHOD

We apply symbolic analysis to nonlinear analog circuits using an automatic simplification procedure. The concept of our approach, is shown in figure 1.

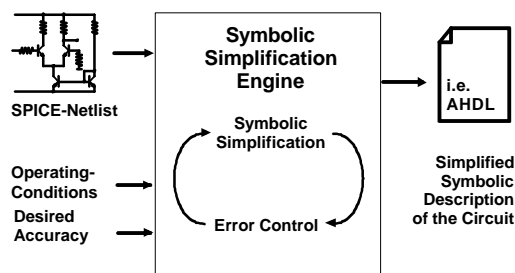


Figure 1: Concept of symbolic simplification

Its core is a simplification engine, which works fully automatic 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. Therefore, a library of appropriate symbolic model equations for all circuit elements has been developed. The created set of DAEs is reduced step by step using various types of symbolic simplification. After each step 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 a testbench in which a circuit is connected for simulation, and by the performed simulations with their ranges. The result is a symbolic description consisting of a new set of equations with a drastically reduced number of equations and terms.

## SIMPLIFICATION OF NESTED EXPRESSIONS

The extensions of this methods within our approach concern the reduction of the set of DAEs not only with respect to the number of equations and terms, but also to the size of terms. Concerning bipolar devices, this has not been necessary, because the largest addends within the model equations have been small enough, so that they could be removed as a whole. For example, consider a large term of the Gummel-Poon model

$$\dots + \frac{\partial}{\partial t} \left( a_3 \cdot \frac{e^{a_4(x_5(t) - x_6(t))} \cdot (x_5(t) - x_6(t))}{a_5} \right) + \dots = 0 \quad (1)$$

in which the  $x_i(t)$  represent circuit variables and the  $a_i$  represent numeric or symbolic constants. For simplification purposes such a term may be removed as a whole within a circuit equation.

Concerning MOS device models, the situation is different. Here, the model equations consist of larger addends, than in the bipolar case and less addends per equation. For example, BSIM3V3 model equations contain terms like:

$$\dots + a_1 \sqrt{1 + \frac{a_2(a_3\sqrt{x_1(t)^2 + a_4x_1(t) + a_5}}{\sqrt{a_6 - a_6x_3(t) + a_6\sqrt{x_1(t)^2}}}} \left( a_7 + a_8 \sqrt{\frac{a_9 - a_6x_1(t) + a_6\sqrt{x_1(t)^2}}{a_{10} - a_4x_1(t) + a_4\sqrt{x_1(t)^2}}} \right) - a_{12}^2 + a_{13} + \dots \quad (2)$$

It can be assumed, that an equation with such an additive term, in general cannot be simplified by removing this term as a whole, because the error would be too large. Especially if this term would represent one side of an equation, it will be impossible to remove it. A promising simplification operation of such equations is to remove nested expressions of those terms.

Therefore, an algorithm has been developed, that enables a reduction of large addends by simplification of its

nested expressions. It builds up a directed, ordered tree of all nested expressions of any term, that shall be simplified. The tree is used to perform the simplification, which depends on the connecting mathematical operation. For example, consider the nominator of the fraction in (2):

$$a_2(a_3\sqrt{x_1(t)^2+a_4x_1(t)+a_5}) \quad (3)$$

It has a nested structure of terms, which can be described by a tree, which is build recursively. In this tree, mathematical operations, like sums or squares and each term with more than one operand represent a node. Numbers, parameters or variables represent the leaves of the tree. The term in (3) leads to a tree, shown in figure 2:

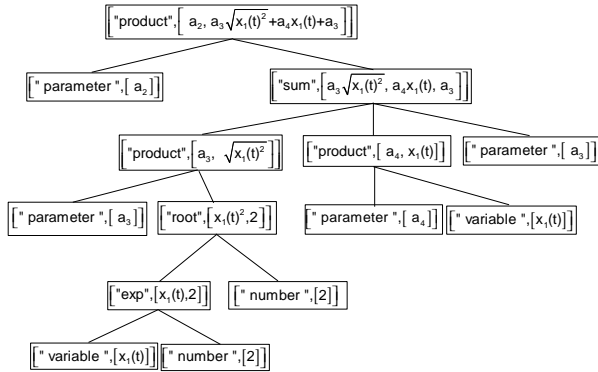


Figure 2: Directed tree of the expression in (3)

With respect to the connecting operations of a term, simplifications are possible. For example, sums can be simplified by removing addends and products become more simple by setting its factors to constant values or even to one. Concerning the method it is not important, whether the constants are numeric or symbolic. The value of those constants can be taken from the user defined operating conditions and simulations respectively.

Concerning powers, roots, logarithms or exponentials, the situation is more difficult. Certainly they could be linearized, but this is no smart simplification, which omits knowledge, that is contained within a circuit or device description. For example the sixth element of the tree above, which is a squareroot of a square. This expression is derived from a device model equation, which represents the absolute value of the basis of the squareroot. The expression can be substituted by its basis, here  $x_1(t)$ , if the basis - under model operating conditions - is always positive.

To realize such case sensitive simplifications, the tree has to be build in advance, before the first simplification is performed. The algorithm is shown in figure 3:

```

build_tree (TERM)
  Analyze operation of TERM
  Concatenate OPERATION and its arguments to tree
  For each argument in TERM do
    build_tree (argument of TERM)
  next argument of tree

```

Figure 3: Recursive algorithm to build the tree of a term

Once the tree has been built, it can be used to decide, which kind of simplification has to be done. Therefore patterns of terms to be simplified specially, have to be

defined with respect to the device model equations. For example as pattern of the root of a square we get:

$$\left[ \begin{array}{l} \text{"root", [(term), 2]} \\ \text{"exp", [(term), 2]} \end{array} \right] \quad (4)$$

On the whole, the process of simplifying nested expressions is divided up into three parts: First, a pure symbolic preprocess to build the tree, to analyze the nested terms and to prepare the simplification. Second the actual error controlled simplification process, and third a symbolic postprocess. Apart from the construction of the tree and the analysis of the possible case sensitive simplifications, the symbolic preprocess consists of another part [8], [9]: It is the insertion of additional factors into the set of DAEs, in order to allow virtual symbolic simplifications in a numerical simulator. For example see the sum contained in (3) after the insertion of some factors:

$$\text{factor}_1 \cdot a_3 \cdot \sqrt{x_1(t)^2} + \text{factor}_2 \cdot a_4 \cdot x_1(t) + \text{factor}_3 \cdot a_5 \quad (5)$$

By setting  $\text{factor}_i$  to zero, the underlying symbolic simplification of the removal of the  $i$ -th term can be performed virtually. Following this principle several symbolic simplifications of various types can be prepared. After such a preparation several simplifications can be checked within one simulator call by a controlled switching of the inserted factors between 0 or 1. In the symbolic postprocess the information, whether a factor can be set to 0 or not leads to the allowed simplifications.

## RESULTS

To show the difference between the capabilities of the simplification with and without simplification of nested expressions, our method has been applied to two circuits: A simple operational amplifier in bipolar technology shown in Figure 4:

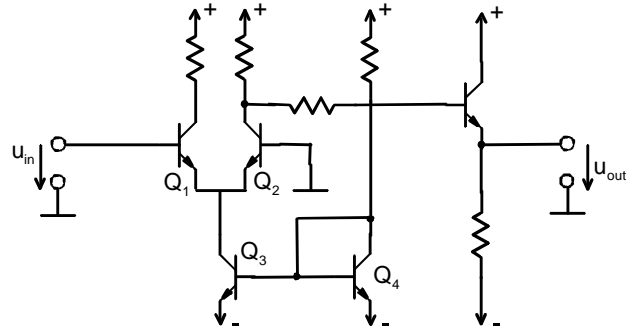


Figure 4: Simple operational amplifier

and an active-loaded common-emitter gainstage [10] in and a BiCMOS technology shown in Figure 5:

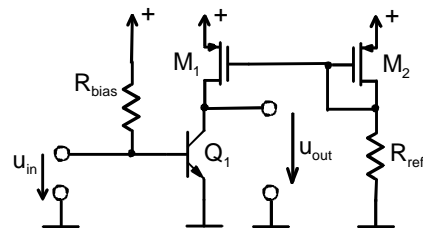


Figure 5: Active-loaded common-emitter gainstage

### A. Results in Bipolar Technology

The simple OP is analyzed in a testbench, which connects two voltage sources as power supply and one as input source, which is connected to the negative input. The positive input is put to ground, whereas the output is connected with a load resistor and a load capacitor in parallel. As simulations, a DC-Transfer and three AC simulations at different operating points are performed.

The acceptable error has been set to 2% for the DC-Transfer and 1% for the amplitude and 10% for the phase of the AC simulations. Figures 6 and 7 show the output signals  $v_{out}(t)$  for the original (orig) and the reduced (red, nested and red, simple) set of DAEs in both simulations:

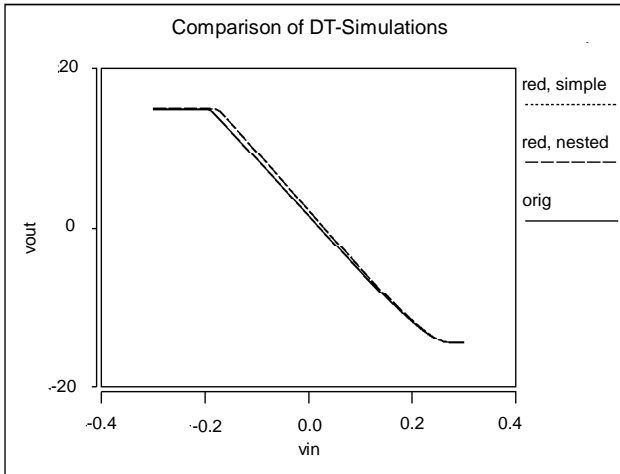


Figure 6: Signal  $v_{out}(v_{in})$  of DC-transfer simulations

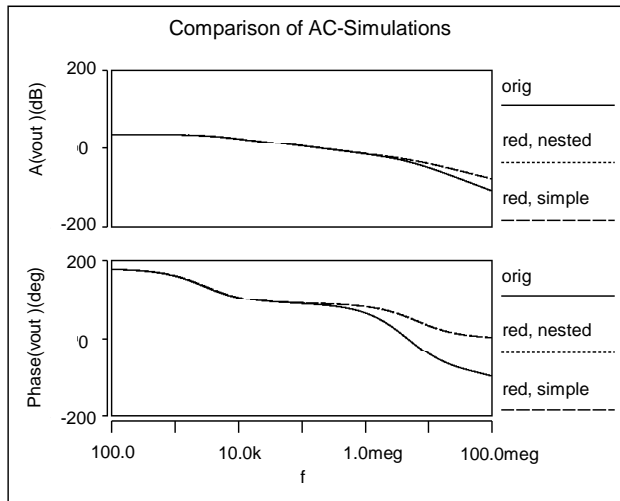


Figure 7: Amplitude and phase of  $V_{out}$  in one operating point

The figures show, that the simplified descriptions of the circuit characterize the input/output behavior within the predefined accuracy. Remarkable is, that the simulations of both simplified descriptions are nearly the same, although the simplification algorithms differ. This concludes, that the simplification ratio of the two algorithms is similar, which has to be verified. In Table 1 the sizes of the sets of DAEs are shown, using three different types of measurement.

The first one counts the number of variables, the second one the number of toplevel addends and the third one the

number of characters of the set of DAEs. All types of measurements allow conclusions about the size of a set of DAEs. Although, none of them is able to measure an objective complexity. Apart from that the number of ASCII characters in a set of DAEs is the one, that considers simplifications of nested expressions. Therefore, we take this type of measurement to compare the results of the different simplification algorithms. Additionally, we keep the others, not to measure the simplification ratio, but to have a rough guess of the size of a set of DAEs.

Number of "..." in the set of DAEs	Original	Simple Simplification	Nested Simplification
Variables	22	13	13
Toplevel Addends	133	30	30
ASCII characters	7109	1117	977

TABLE 1: Simplification Results on the bipolar OP with and without Simplification of Nested Expressions

Table 1 shows that the reduction ratio of the simplification engine increases from factor 6,4 to factor 7,3, by using the simplification algorithm of nested expressions. This small improvement has had to be expected, because of the structure of terms in bipolar model equations mentioned above. As expected, the number of variables and toplevel addends do not differ

### B. Results in BiCMOS Technology

The operating conditions of the common-emitter-gainstage have been fixed by two simulations, a DC-Transfer and a transient simulation. To perform these simulations the gainstage has been connected to a testbench, containing voltage sources for power supply and for input purposes and two resistors dealing as input series resistor and as load resistor, respectively.

The acceptable error has been set to 10% for the DC-Transfer and 20% for the transient simulation.

Figures 8 and 9 show the output signals  $v_{out}(v_{in})$  and  $v_{out}(t)$  for the original (orig) and the two reduced (red, nested and red, simple) set of DAEs in both simulations:

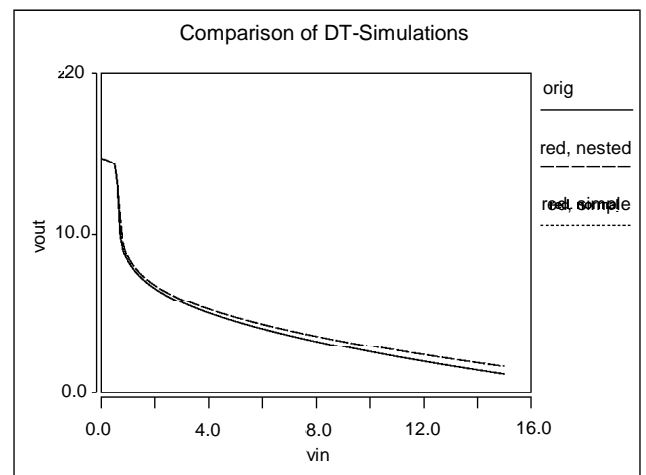


Figure 8: Signal  $v_{out}$  of DC-Transfer simulation

Concerning the DT-simulation results, the input/output behavior of the simplified descriptions are nearly equal – as in the bipolar case – and give a good approximation of

the original circuit's behavior. The similarity is surprising, because of the achieved different simplification ratios of the two algorithms, which are shown in Table 2 below.

On the contrary, the transient simulations show three different input/output behaviors, according to three different circuit descriptions.

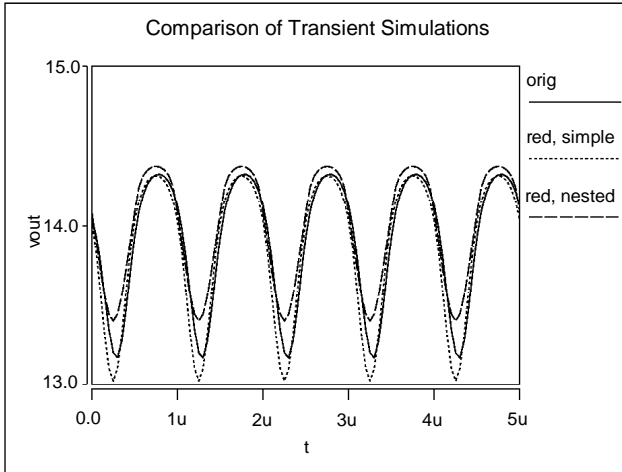


Figure 9: Signal  $v_{out}(t)$  of transient simulation

Due to the error controlled algorithm, both simplified descriptions lead to signals within the desired accuracy. In Table 2 the simplification results are shown:

Number of "..." in the set of DAEs	Original	Simple Simplification	Nested Simplification
Variables	78	29	18
Toplevel Addends	293	68	45
ASCII characters	25372	8757	3391

TABLE 2: Simplification Results on the BICMOS Gainstage with and without Simplification of Nested Expressions

Remarkable is, that despite our expectation, a reduction ratio can be measured with all proposed types of measurement. It varies between 2,7 and 4,3 in the simple and 4,3 and 7,5 in the nested case. The unexpected reduction ratio in the number of variables and toplevel addends results of substitutions, which have been possible by the simplification of large nested terms. Because of its dependency of particular circuits, both ratios are not well suited for measurement of the reduction ratio in general. Therefore we concentrate on the number of ASCII characters.

The reduction ratio of the simplification engine increases from factor 2,9 to factor 7,5 by using the simplification algorithm of nested expressions. This is a quite good result, if it is taken into account, that one out of three transistors is a bipolar one, which can be treated by the simple algorithm.

## CONCLUSION

A method for symbolic analysis of strongly nonlinear analog circuits including MOS devices has been presented. It is based on symbolic simplification of nested expressions in a set of nonlinear differential algebraic

equations (DAEs). The feasibility of the method has been shown using circuit examples, pointing out the difference between bipolar and MOS devices.

## ACKNOWLEDGEMENT

This work has been supported within a subcontract between the TEMIC Semiconductor GmbH and the IMS. This contract is part of the MEDEA Project "Systematic Analog Design Environment", (SADE).

## REFERENCES

- [1] G. Gielen, W. Sansen, "Symbolic Analysis for Automated Design of Analog Integrated Circuits", Kluwer Ac. Pub., 1991
- [2] P.-M. Lin, "Symbolic Network Analysis", Elsevier, 1991
- [3] F.V. Fernandez, A. Rodriguez-Vazquez, "Symbolic Analysis Tools - The State-of-the-Art", Proc. IEEE Int. Symposium on Circuits and systems, p.798-801, 1996
- [4] 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
- [5] C. Borchers, L. Hedrich, E. Barke, "Equation-Based Behavioral Model Generation for Nonlinear Analog Circuits", Proc. DAC, pp. 236-239, 1996
- [6] R. Popp, W. Hartong, L. Hedrich, E. Barke, "Error Estimation on Symbolic Behavioral Models of Nonlinear Analog Circuits", SMACD'98, pp. 223-226, Kaiserslautern, 1998
- [7] 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
- [8] R. Popp, L. Nathke, C. Borchers "Automatische Erzeugung symbolischer Verhaltensmodelle fur nichtlineare Anlogschaltungen im transienten Grosignalbetrieb", 5. GMM/ITG Diskussionssitzung ANALOG 99, Munchen, 1999 (German Language)
- [9] L. Nathke, R. Popp, L. Hedrich, E. Barke "Using Term Ordering to Improve Symbolic Behavioral Model Generation of Nonlinear Dynamic Analog Circuits", Proc. ECCTD'99, pp. 74-77, Stresa, Italy, 1999
- [10] E. W. Greneich, "Analog Integrated Circuits", Chapman & Hall, 1997