

Symbolic Modeling and Analysis of Analog Integrated Circuits

Ralf SOMMER¹, Eckhard HENNIG¹, Manfred THOLE¹, Thomas HALFMANN¹, Tim WICHMANN¹

ABSTRACT

In this paper an overview of the application of symbolic analysis and computer algebra is given. After an introduction requirements to symbolic analysis tools are formulated, and a short abstract of a general symbolic equation-based approximation algorithm is given. A generic symbolic analysis flow is introduced and applied to derive a nonlinear behavioral model of an integrated multiplier circuit using the symbolic analysis toolbox Analog Insydes [1]. It will be shown how symbolic analysis can assist behavioral model generation and can help to better understand a circuit in order to improve the quality of the design.

1. INTRODUCTION

Analog and mixed-signal design is of great importance in microelectronics applications, like automotive and telecommunication. The traditional design of analog integrated circuits relies largely on a mixture of expertise, some manual calculations, and numerical circuit simulation. Recent research and development in the field of symbolic circuit analysis has produced results which may have considerable impact on some parts of the traditional design flow, but few analog designers have adopted symbolic analysis techniques as standard tools in their CAD environments yet. To a large extent this may be due to the lack of documented methodologies which show what can be expected from symbolic analysis and how it can be efficiently employed to solve industrial circuit design problems. A tool assisting analog expert designers in circuit sizing, optimization, and characterization is now urgently needed to enhance design productivity in order to face shrinking time-to-market schedules.

The application fields of symbolic analysis techniques (in a close connection with numerical methods) can be divided into the following four main categories, which are essential tasks in the industrial design flow of analog integrated circuits:

Circuit analysis:

- determine the influences of element parameters on circuit behavior

- extraction of dominant circuit behavior in a mathematical and interpretable form (also to be used for circuit sizing)
- error and tolerance analysis

Circuit modeling:

- support of model generation for analog circuit blocks (on different hierarchical levels)
- allow for overall circuit simulation by use of behavioral and macro-models

Circuit sizing:

- support manual or computer-aided circuit synthesis
- derivation of symbolic (generic) sizing formulas for circuit elements as functions of global circuit specifications

Circuit optimization:

- preprocessing of equations by e.g. elimination of variables to allow for an efficient optimization run
- allow for application of optimization algorithms already on system level

2. REQUIREMENTS FOR SYMBOLIC ANALYSIS TOOLS

As a consequence of the large variety of application fields summarized in the previous section it becomes apparent that state-of-the-art symbolic analysis tools have to be characterized by flexibility in their functionality as well as transparency in their data structures and models. Moreover comfortable interfaces to the user on the one hand and to numerical simulation environments on the other hand must be provided because symbolic analysis is no stand-alone application any more and has to be embedded into the designer's workflow. The following key requirements were identified in many technical discussions with circuit designers.

Equation formulation: To provide flexibility in analysis modes as well as to assist a designer in model development symbolic analysis tools should allow for setting up circuit equations not only for linear circuits in the frequency domain but also in the time domain for both –linear and nonlinear– circuits and systems. For modeling purposes and for better interpretability of expressions equation formulation should not be restricted to special types of elements (conductances) or circuit analysis representations (e.g. MNA).

¹ ITWM – Institute of Industrial Mathematics, Kaiserslautern, Germany

Hierarchy: Since most analog circuits are designed following a hierarchical approach a symbolic analysis tool must allow for hierarchical circuit description in terms of circuits, subcircuits and device models, and must provide support for specification mapping and propagation of parameters between hierarchy levels. In addition circuit data representation must support the parallel implementation of different abstraction levels for a circuit block. Such partial abstractions and computations with mixed hierarchy levels are just as important for symbolic circuit analysis as for numerical simulation. The underlying idea is to replace the surrounding circuitry by a simpler behavioral description of its input/output characteristics while only the block under test is simulated at the device level.

Device modeling: Careful modeling of devices is one of the main prerequisites for successful application of symbolic circuit analysis. Failure to choose simple models generally results in extremely large expressions which cannot be interpreted or even computed at all. Depending on its individual function, each device in a circuit should be modeled in the simplest possible way whose impact on overall simulation accuracy is still tolerable. This requires application-specific and even instance-specific device modeling.

Determining the best compromise between model accuracy and expression complexity is often an iterative process in which various models must be tried for a device until a satisfactory analysis result is obtained. Selecting and exchanging device models must therefore be quick and easy, and should not involve tedious netlist editing operations.

Data integration and interfaces: Circuit representation must fully integrate all symbolic and numerical data which is necessary for model definition and expansion, parameter translation and propagation, symbolic approximation, etc. Since symbolic methods are hardly ever applied independently of numerical circuit simulation, simulation results, such as operating point data and small-signal parameters, are always required as input for symbolic approximation routines. Moreover, symbolic analysis results should always be verified against numerical simulation so that processing simulator output data is an additional feature to reading in netlists, model cards and operating-point information.

3. SYMBOLIC APPROXIMATION STRATEGIES

Practical application of symbolic analysis would have been rather limited without application of symbolic approximation techniques. Indeed these techniques hold the key in modern symbolic circuit analysis. A lot of research has been done and reported in this area resulting in three different categories of approximation strategies: Simplification after generation (SAG), Sim-

plification during generation (SDG), and Simplification before generation (SBG).

One of the central prerequisites of the symbolic analysis flow presented in the next section was the development and implementation of efficient symbolic approximation algorithms which impose no restrictions on the formulation of circuit equations, neither linear nor nonlinear, or the set of circuit elements that may be used.

Equation-based approximation procedures own all these requested properties since they are already applied on the level of circuit equations before the solution is determined (SBG). The philosophy behind equation-based approximation is to follow the methodology of a circuit designer who introduces his simplifications already when formulating equations. Thus the complexity of the problem and the mathematical effort to solve or process the system is reduced substantially.

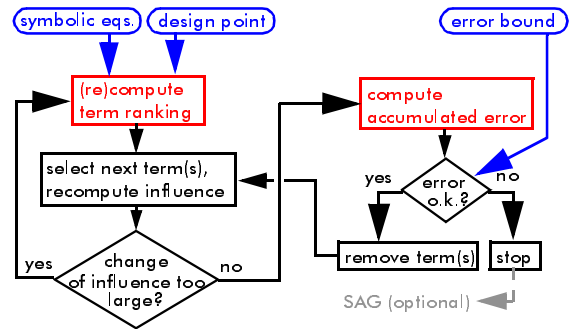


Figure 1: Flow of equation-based approximation

Since this paper intends to give an overview of methodologies and results, only the underlying principle of equation-based approximation is presented. Figure 1 shows a general flow chart of the algorithm.

Equation-based approximation starts with the system of symbolic linear or nonlinear equations and a list of corresponding numerical reference values called design point.

Based on these numerical reference values the system of symbolic equations is evaluated and solved. This information is subsequently used to generate a term ranking. The term ranking mechanism plays a key role in the algorithm. Its task is to compute an order of all symbolic terms of the underlying equations such that the terms are sorted with respect to their influence on the solution. Ranking algorithms are an important subject of research since a large variety of different circuit characteristics may be of interest which have to be taken into account by the algorithm. For example in linear analysis magnitude, phase as well as pole and zero locations are of interest while in nonlinear analysis DC transfer, transient behavior, distortion, etc. are to be captured by the approximated system.

In the next step the output of the ranking algorithm is

processed by the term removal mechanism which removes one or more terms from the system of symbolic equations. Now this manipulated system with one or more terms deleted is passed to the error checking routine. Here the accumulated numerical error caused by the term removal is calculated and compared with the given error bound. If the error bound is exceeded the last term removal is undone and the algorithm terminates returning the approximated system. If the error bound is not exceeded the next term or terms from the term ranking list are selected and removed from the system followed by the error checking procedure as already described before. There are several extensions to the algorithm, e.g. symbolic simplification and elimination steps as well as more sophisticated term removal operations, e.g. block removals of elements and to use the error checking routine to control the term ranking [3].

4. SYMBOLIC ANALYSIS WORK FLOW

In this section a modeling strategy for the derivation of approximated symbolic expressions for selected circuit characteristics is presented. This strategy has already successfully been applied to the derivation of industrial-sized linear circuits [2]. A particularly important aspect of this approach is the interaction between symbolic and numerical computations (e.g. SPICE) to ensure continuous error control and verification of the results. The general flow (Figure 2) can be divided into the following steps:

1. Start with a numerical SPICE simulation of the circuit under examination and make sure that the effect of interest can be observed.
2. Focus on one task or effect of interest and use the mathematically simplest analysis method (e.g. no transient analysis for a small-signal effect)
3. Generate a symbolic netlist with numerical reference information for semi-symbolic analyses and symbolic approximation techniques
4. Select the same set of device models for symbolic analysis as for the preceding numerical analyses in Step 3.
5. Ensure validity of netlist and models by comparing semi-symbolic analysis results with the numerical simulation.
6. Iteratively select simpler device models as long as the deviation from the SPICE reference simulation is tolerable. Check deviations by semi-symbolic analysis without employing approximation methods.
7. Perform symbolic analysis using symbolic approximation, pole/zero extraction, etc. Always check numerically evaluated results against the reference simulation.
8. For any result (numeric and symbolic): Perform plausibility check.

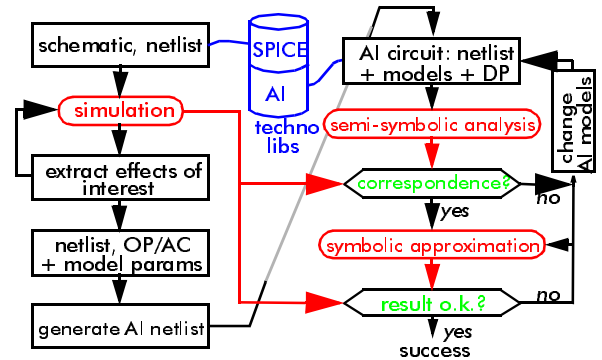


Figure 2: Symbolic analysis work flow

5. EXAMPLE: DERIVATION OF A BEHAVIORAL MODEL OF A MULTIPLIER CIRCUIT

As an example a multiplier circuit (Figure 3) is analyzed according to the flow diagram using Analog Insydes. Symbolic analysis is applied to extract an approximated formula which describes the nonlinear DC transfer characteristic in terms of the circuit parameters. Following some excerpts from the whole procedure are presented.

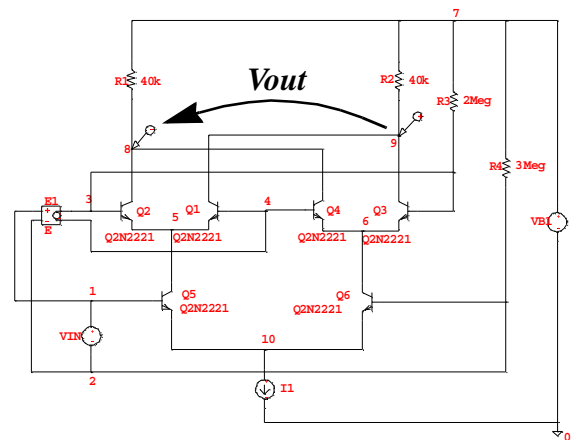


Figure 3: Schematic of multiplier circuit

Steps 1-6: We start with a PSpice DC transfer analysis of the circuit yielding the characteristic shown in Figure 6. Next all data, i.e. netlist, BJT parameters, and simulation data are read into Analog Insydes. A semi-symbolic analysis is performed using the full Gummel-Poon BJT model yielding a nonlinear system of 68 equations with 265 terms. The numerical solution of these equations is identical to the PSpice simulation so in step 6 the simplified Ebers-Moll BJT model is chosen yielding the 26×26 system of equations with 118 terms shown in Figure 4. Since a numerical simulation shows again no deviation to the original PSpice simulation we proceed with step 7. The equation-based ap-

