SYMBOLIC ANALYSIS TOOLS - THE STATE-OF-THE-ART

Francisco V. Fernández and Angel Rodríguez-Vázquez

Dept. of Analog Circuit Design, Centro Nacional de Microelectrónica
Edif. CICA, Avda. Reina Mercedes s/n, E-41012 Sevilla, SPAIN
Phone: +34 5 4239923, FAX: +34 5 4231832, E-mail: pacov@cnm.us.es

ABSTRACT

This paper reviews the main last generation symbolic analyzers, comparing them in terms of functionality, pointing out also their shortcomings. The state-of-the-art in this field is also studied, pointing out directions for future research.

1. INTRODUCTION

Circuit analysis is a basic milestone for efficient design of integrated circuits. Ever since powerful computers have been available, designers have developed programs to analyze circuits automatically. Today, all electrical engineering professionals and students use *electrical simulators* (such as HSPICE or ELDO). However, electrical simulators do not cover all the analysis tasks required for integrated circuit design. Essentially, they only serve to *verify* the performance of previously *sized* circuits. Among other things, designers must be able to *predict* the behavior of *unsized* circuits by tracing relationships among performance figures and design parameters. These relationships may be in the form of transfer functions, poles and zeros, root loci, parametric Bode plots, harmonic distortion coefficients, etc. Tools used to derive them automatically are called *symbolic analyzers*.

The first generation of symbolic analyzers appeared in the late 60´s [1], where the basic analysis techniques were proposed. Most of these techniques are compiled in a recent book by P.M. Lin [2]. These tools did not spread on the designer community, partially due to their computational cost and because they did not fully fit the designer requirements. On the other hand, the lack of interest on custom analog ICs during the 70´s rendered symbolic analyzers secondary as compared to electrical simulations. This scenario changed during the 80´s with the increased economical interest on custom analog and mixed-signal ASICs. Symbolic analyzers were promptly recognized crucial tools to: provide insight and knowledge about analog circuits; support automatic behavioral modeling of analog integrated circuits; provide compiled models for automatic repetitive evaluations required for analog synthesis, among other applications [3], [4]. It resulted in the emergence of a new generation of tools during the late 80´s. Main features of this new generation have been approximation, post-processing and the extension to weakly non-linear circuit analysis. This paper summarizes some main contributions in these second generation tools, and compares the characteristics of many of them.

2. SYMBOLIC EXPRESSION APPROXIMATION

Simplification in symbolic analysis is defined as the reduction of formula complexity by eliminating insignificant terms or sub-expressions, based on numerical estimates of the symbolic parameters. Simplification (or approximation) is required even for small circuits given the exponential increase of the expression length with the circuit size. Conventional simplification approaches first calculate the *complete* symbolic expression, and then simplifies it -- Simplification After Generation (SAG). Different versions of SAG have been implemented in modern symbolic analyzers: ISAAC [5], SSPICE [6] and ASAP [7] for *expanded* format expressions, and SYNAP [8] and ASAP [7] for *nested* format expressions. Differences arise in the control of errors and in the use of either a nominal point or a nominal interval to estimate the magnitude of the symbolic terms.

Due to the necessity of generating complete expressions, SAG techniques are constrained to small and medium complexity circuits (below about 100 symbols). Larger circuits can be analyzed by using the newest approaches of simplifying *during*, or *before* to, the analysis stage. They are called Simplification During Generation (SDG) and Simplification Before Generation (SBG) techniques respectively.

The rationale behind SDG techniques is that terms can be generated strictly in decreasing order of magnitude starting with the largest. The classical undirected *tree enumeration* method has demonstrated to be the most efficient to solve the underlying *matroid* intersection problem. This approach has been implemented with slight differences in RAINIER [9],[10] and ADAGIO [11],[12], the new tool evolved from ASAP and ISAAC.

SBG can take place in the network under analysis, replacing those elements (or subcircuits), whose contribution (appropriately measured) to the network function is negligible, with a zero-admittance or zero-impedance element. These approximations are usually done when deriving network equations manually. However, except for the most trivial replacements, no automatic procedure has been reported yet. Reported approaches perform approximations directly on the network equations, either in graph [9] or matrix form [13], [14].

3. POSTPROCESSING TOOLS

These are basically intended for easier interpretation and manipulations of symbolic formulae, or to extract information from them.

3.1. Symbolic pole-zero extraction

Poles and zeros are commonly used by analog designers. [15] introduced a technique for symbolic extraction of poles and zeros, based on heuristic approximations, similar to those performed by expert analog designers, e.g.: the *root-splitting* technique. This approach is also used in [16] which subsequently applies a symbolic Newton iteration to refine the pole/zero locations obtained with the root-splitting technique.

Consider, for example, the active feedforward compensated amplifier of Fig. 1. A symbolic extraction of the first pole/zero pair gives [15]:

0-7803-3073-0/96/$5.00 ©1996 IEEE

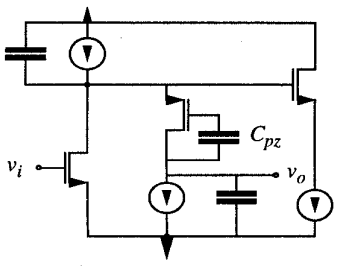

Figure 1: Active feedforward compensated amplifier

$$z_1 = -\frac{g_{ma}}{C_{pz}+C_{gd2}}$$
$$p_2 = -\frac{g_{ma}(C_L+C_{gd2}+C_{pz})}{(C_{pz}+C_{gd2})(C_L+C_{db1}+C_{gd1}+C_{l1})} \quad (1)$$

The interest of symbolic poles and zeros can be seen through this example. From (1) one can infers the value of the compensation capacitor that annuls the pole/zero mismatch and hence improves the high frequency behavior of the amplifier,

$$C_{pz} \cong C_{db1}+C_{gd1}+C_{l1}-C_{gd2} \quad (2)$$

Another extraction technique, reported in [17], extends the root splitting technique to cluster of poles or zeros which are quite apart from other clusters and calculates approximate expressions of the network function for a frequency range appropriate to the roots being calculated.

3.2. Graphical interfaces

Some tools incorporate graphical interfaces for parametric representations of Bode diagrams or root locus. For illustration, Fig. 2 shows the graphical representation of the real part of poles and zeros of the voltage gain of Fig. 1 versus the value of C_{pz}. The value for which the pole/zero pair cancel is clearly seen.

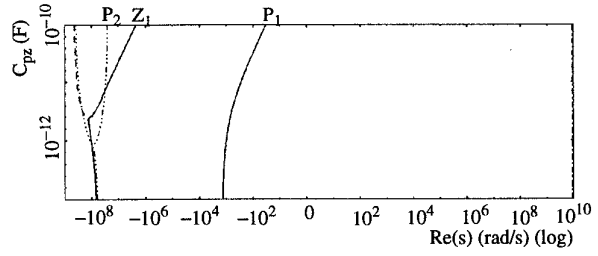

Figure 2: Pole/zero location (real part) versus C_{pz}.

3.3. Sensitivity analysis

Sensitivity analysis also plays a fundamental role in analog circuit design, specially in the calculation of the influence of component mismatches. The sensitivity of a circuit characterized by a symbolic system function $H(s, x)$ can be obtained by simply deriving the system function with respect to the circuit parameter [18], [19], [20].

3.4. Factorization and complexity reduction

Even approximated expressions can be difficult to interpret due to the considerable size they have when calculated with enough accuracy in medium or large circuits. Factorization of common factors and cancellations of common factors in numerator and denominator are some of the needed capabilities to improve the interpretability without decreasing the accuracy. Many modern tools incorporate some primitive form of factorization. But more sophisticated mechanisms are still needed, for instance to eliminate the contribution of bias circuitry which usually affect equally numerator and denominator of most transfer functions.

4. AN OVERVIEW OF CURRENT TOOLS

Many symbolic analyzers have evolved as members of this second generation. Exact comparison of their performance is difficult due to the lack of standard benchmarks for test circuits and platforms. Some key characteristics of main reported symbolic analyzers are compared in Table 1. A brief description of those analyzers follows.

SCAPP [22] has been implemented at Iowa State University, in the framework of the development of a new methodology for symbolic analysis of large circuits. The program partitions the circuits hierarchically and symbolically analyzes each sub-circuit separately. Once the terminal blocks are analyzed, an inverse run of the partition process reconstructs the global system function. Since this analyzer uses an approach for large circuits based on hierarchical analysis, the system functions are given as a series of expressions rather than only one expression. Thus, the number of symbolic terms grows linearly, instead of exponentially, with the circuit size. SCAPP can be obtained for free via anonymous ftp at cpre1.ee.iastate.edu:/pub/scapp

SCYMBAL and SYBILIN [23] are two simulators developed in CNET (Bagneux, France). A distinctive feature of SCYMBAL is that it was developed specifically for switched-capacitor circuits. It allows not only continuous variables such as conductances, capacitors, etc., as symbolic parameters, but also boolean variables such as clock phases. On the other hand, SYBILIN is a continuous-time analyzer, specially conceived for the analysis of microwave circuits. These tools can be obtained from the authors on request. The price depends on the type of institution.

SC [24] was developed at the University of Arizona, and conceived for classroom use, and thus developed for personal computers. It has a set of post-processing tools to ease interpretation of symbolic data: interactive numeric evaluation, numeric calculation of poles and zeros, and graphic representation. SC can be obtained from the authors with a $6 handling charge.

SAPEC [25] was implemented at the University of Florence, also for PCs. One of its main characteristics is the group of applications developed around it, such as symbolic sensitivity analysis, transient analysis of electronic power circuits, calculation of testability index and fault diagnosis. SAPEC can be obtained for free from the author (e-mail: manetti@ingfi1.ing.unifi.it)

SSPICE [6] was developed at Michigan State University for both personal computers and mainframes. It incorporates a fast albeit not very accurate SAG technique. It is also capable of performing sensitivity analysis [19]. SSPICE is available from Instructional Media Center (Michigan State

University) with a cost of $500/copy and a license fee of $250/year. There is a limited student version for $25.

SYNAP [8] was developed at ETH in Zurich (Switzerland). Together with ISAAC and ASAP, it was especially conceived for the modelling and design of analog integrated circuits and, more specifically, to be included on CAD frameworks for automatic design of analog integrated circuits. It incorporates encapsulated models for semiconductor devices and analog functional blocks, and can handle mismatches. Its newest version is capable of analyzing linear continuous-time and sample-data circuits, and also incorporates a primitive DC analysis capability. SAG techniques implemented in SYNAP use the concept of "lazy expansions", which performs approximations directly on the nested expressions. SYNAP is available from the author only to universities and non-profit research institutions.

The first version of ISAAC [5] was developed in the Katholieke Universiteit Leuven. It was the first tool to incorporate the analysis of weakly nonlinear characteristics [26]. Like ASAP, SYNAP, and SSPICE it obtained simplified expressions using SAG techniques. It uses exact expressions in expanded format and eliminates those least significant terms whose accumulated sum remains small. ISAAC forms a fundamental part of the automatic design system ARIADNE [27]. ISAAC is available from the authors for a $400 handling charge. A Lucid Common Lisp license is required. Distribution to companies is done on a case by case basis. Currently it is being improved to include SDG and SBG, among other new features -- a cooperative project shared with ASAP.

ASAP [28] was developed at the Centro Nacional de Microelectrónica in Sevilla (Spain). Like SSPICE, it runs on workstations and PCs. It shares with ISAAC the capability of performing groupings of elements and together with SYNAP, the explicit expression of mismatching. It calculates approximate poles and zeros symbolically. ASAP also incorporates a new SAG criteria which reduce errors by considering intervals of the symbolic parameters instead of fixed numeric values. Approximation with variation ranges has also been used to develop a reliable and precise criterion to simplify nested expressions resulting from hierarchical analysis. ASAP has been commercialized within the ELDO framework and is available through ANACAD-EES.

SIFTER [14] and RAINIER [9] have been developed at the University of Washington at Seattle, and have mainly addressed the implementation of SBG and SDG techniques, as discussed in Section 2. RAINIER is available for free via anonymous ftp at twolf8.ee.washington.edu.

5. FUTURE DEVELOPMENTS

One of the major challenges for future symbolic analyzers is increasing the complexity of circuits, while keeping the interpretability of results. A possible solution has been shown to be the application of hierarchical analysis techniques. These techniques are specially adequate for large circuits consisting of loosely coupled sub-blocks. For strongly connected circuits new SBG and SDG techniques are arising. The symbolic analysis of large networks has been the focus of intense attention in the last years, is still of interest today, and doubtless will continue to be of interest in the future.

Table 1 shows that most tools are limited to the analysis of linear(ized) circuits. The first attempt to nonlinear analysis has been reported in [26] although limited to weak non-linearities. Reducing the actual symbolic analysis techniques to the study of linear or weakly nonlinear characteristics limits its practical applications. It can be assured that the possible future availability of large signal symbolic analysis techniques will suppose the full integration of these tools in automatic design and test systems, and in general, within any CAD environment. Regarding the time domain, the first attempts at time-domain symbolic simulation have been reported in [29].

Together with the development of symbolic analysis techniques for non-traditional characteristics, and the new, more efficient algorithms, applications are being amplified within fields such as automatic design of analog integrated circuits.

REFERENCES

[1] P. M. Lin, "A Survey of Applications of Symbolic Network Functions," *IEEE Trans. on Circuit Theory*, Vol. CT-20, pp. 732-737, Nov. 1973.

[2] P.M. Lin, *Symbolic Network Analysis*. Elsevier, 1991.

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

[4] A. Rodríguez-Vázquez et al. (eds.), *Analog CAD Tools: Symbolic Techniques and Applications*. IEEE Press, 1996.

[5] G. Gielen, H. Walscharts and W. Sansen, "ISAAC: A Symbolic Simulator for Analog Integrated Circuits," *IEEE J. Solid-State Circ.*, Vol. 24, No. 6, pp. 1587-1597, Dec. 1989.

[6] G. Wierzba, A. Srivastaba, V. Joshi and K.V. Noren, "Sspice - A Symbolic SPICE Program for Linear Active Circuits," *Proc. IEEE Midwest Symp. Circuits and Systems*, pp. 1197-1201, 1989.

[7] F.V. Fernández, A. Rodríguez-Vázquez, J.D. Martín and J.L. Huertas, "Formula Approximation for Flat and Hierarchical Symbolic Analysis," *Analog Integrated Circuits and Signal Processing*, Vol.3, pp. 43-58, Kluwer, Jan. 1993.

[8] S. Seda, M. Degrauwe and W. Fichtner, "Lazy-Expansion Symbolic Expression Approximation in SYNAP," *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 310-317, 1992.

[9] Q. Yu and C. Sechen, "Approximate Symbolic Analysis of Large Analog Integrated Circuits," *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 664-671, 1994.

[10]Q. Yu and C. Sechen, "Efficient Approximation of Symbolic Network Functions Using Matroid Intersection Algorithms," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 2088-2091, 1995.

[11]F.V. Fernández, P. Wambacq, G. Gielen, A. Rodríguez-Vázquez and W. Sansen, "Symbolic Analysis of Large Analog Integrated Circuits by Approximation During Expression Generation," *Proc. IEEE Int. Symp. Circuits and Systems*, pp.25-28, 1994.

[12]P. Wambacq, F.V. Fernández, G. Gielen, W. Sansen and A. Rodríguez-Vázquez, "Efficient Symbolic Computation of Approximated Small-Signal Charac-teristics of Analog Integrated Circuits," *IEEE J. Solid-State Circ.*, Vol. 30, No. 3, pp. 327-330, Mar. 1995.

[13]R. Sommer E. Hennig, G. Droge and E.-H. Horneber, "Equation-Based Symbolic Approximation by Matrix Reduction with Quantitative Error Prediction," *Alta Frequenza*, Vol. 5, No.6, pp. 317-325, AIE, Nov. 1993.

[14]Jer-Jaw Hsu and C. Sechen, "Fully Symbolic Analysis of Large Analog Integrated Circuits," *Proc. IEEE CICC*, pp. 457-460, 1994.

[15]F.V. Fernández, A. Rodríguez-Vázquez and J.L. Huertas, "A Tool for Symbolic Analysis of Analog Integrated Circuits Including Pole/Zero Extraction," *Proc. European Conf. Circuit Theory and Design*, pp. 752-761, 1991.

TABLE 1: COMPARISON OF SYMBOLIC ANALYZERS

	SCAPP	SCYMBAL /SYBILIN	SC	SAPEC	SSPICE	SYNAP	ISAAC	ASAP	SIFTER	RAINIER
Formulation	RMNA & SFG	SFG	MNA	MNA	MNA	MNA	CMNA	SFG	nodal analysis	tree enum.
Analysis domain	s	z / s	s	s	s	s & z	s & z	s	s	s
SAG (expanded format)	no	no	no	no	yes	yes	yes	yes	no	no
SAG (nested format)	no	no	no	no	no	yes	no	yes	no	no
SBG	no	no	no	no	no	no	no	no	yes	yes
SDG	no	no	no	no	yes	no	yes	yes	no	yes
Mismatchings	no	no	no	no	no	yes	yes	yes	no	no
Element lumping	no	no	no	no	no	no	yes	yes	no	no
Nonlinear analysis	no	no	no	no	no	no	weakly nonlin.	no	no	no
Hierarchical analysis	yes	no	no	no	no	no	no	no	no	no
P/Z extraction	no	no	no	no	no	no	no	yes	yes	no
Graphical interface	no	no	yes	yes	no	no	no	yes	no	no
Platforms (WS ≡ workstations)	WS	WS	WS	PC	WS & PC	WS	WS	WS & PC	WS	WS
Implementation language	C	FORTRAN /ADA	C	LISP/ C++	C	LISP/ C++	LISP/ C	C/C++	C	C

[16]G. Nebel et al., "Symbolic Pole/Zero Calculation Using SANTAFE," *Proc. European Solid-State Circuits Conf,* pp.152-155, 1994.

[17]J. Hsu and C. Sechen, "Accurate Extraction of Simplified Symbolic Pole/Zero Expressions for Large Analog IC's," *Proc. IEEE Int. Symp. Circuits and Systems*, pp.2083-2087, 1995.

[18]A. Liberatore and S. Manetti, "Network Sensitivity Analysis Via Symbolic Formulation," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 705-708, 1989.

[19]S. Chang and G. Wierzba, "Symbolic Sensitivity Analysis Using Sspice," *Proc. IEEE Midwest CAS*, pp. 1043-1046, 1991.

[20]P.M. Lin, "Sensitivity Analysis of Large Linear Networks Using Symbolic Programs," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1145-1148, 1992.

[21]J.A. Echtenkamp, M.M. Hassoun, G.M. Prabhu and C.T. wright, "Hierarchical Sensitivity Analysis for Sequence of Expressions Method," *Proc. European Conf. Circuit Theory and Design*, pp. 75-78, 1995.

[22]M. M. Hassoun and P. M. Lin, "A New Network Approach to Symbolic Simulation of Large-Scale Networks," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 806-809, 1989.

[23]A. Konczykowska and M. Bon, "Symbolic Simulation for Efficient Repetitive Analysis and Artificial Intelligence Techniques in C.A.D.," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 802-805, 1989.

[24]L.P. Huelsman, "Personal Computer Symbolic Analysis Programs for Undergraduate Engineering Courses," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 798-801, 1989.

[25]S. Manetti, "New Approach to Automatic Symbolic Analysis of Electric Circuits," *IEE Proceedings Pt. G*, Vol. 138, No. 1, pp. 22-28, Feb. 1991.

[26]P. Wambacq, G. Gielen and W. Sansen, "Symbolic Simulation of Harmonic Distortion in Analog Integrated Circuits with Weak Nonlinearities," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 536-539, 1990.

[27]K. Swings and W. Sansen, "ARIADNE: A Constraint-Based Approach to Computer-Aided Synthesis and Modeling of Analog Integrated Circuits," *Analog Integrated Circuits and Signal Processing*, Vol. 3, pp. 197-215, May 1993.

[28]F. V. Fernández, A. Rodríguez-Vázquez and J.L. Huertas, "Interactive AC Modeling and Characterization of Analog Circuits via Symbolic Analysis," *Analog Integrated Circuits and Signal Processing*, Vol. 1, pp. 183-208, Kluwer, Nov. 1991.

[29]M. Hassoun et al., "A State-Variable Approach to Symbolic Circuit Simulation in the Time Domain," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 682-685, 1992.