FREQUENCY DOMAIN ELECTRONIC CIRCUIT ANALYSIS IN AN OBJECT ORIENTED ENVIRONMENT

Milan Savić and Vančo Litovski
Department of Electronics, Faculty of Electronic Engineering, Beogradska 14, 18000 Niš
{msavic;vanco}@venus.elfak.ni.ac.yu

Abstract. An extension to the Alecsis (Analogue and Logic Electronic SImulation System) simulator is described in this paper. It is intended to perform frequency domain analysis within the already built object oriented environment. Simulation examples will show the state of the development.

1. INTRODUCTION

Modern electronic simulators mostly use HDL (Hardware Description Language) to establish communication between designer and computer code performing simulation. This is the case with Alecsis [1], too. Namely, Alecsis communicates with the user by the AleC++ language being a superset of C++ [2] as shown in Fig.1. The simulator itself is performing time domain behavioural simulation of mixed signal and mixed domain circuits and systems. It was successfully applied to simulation of a broad set of systems starting with non-linear electronic via digital electronic, trough mixed mode electronic and mixed mode electromechanical, ending with mixed mode behavioural. Now frequency domain analysis is being added.

In addition to features in accordance with IEEE standard for VHDL and VHDL-AMS, AleC++ has features that are very useful for modeling complex systems. Such features for analog modeling are: use of processes in analog models (before defining the topology, when simulator starts, before and after solving the system in every time-instant, in every iteration and at the end of simulation), modifying contributions to connected sub models, which is a very user-friendly modeling paradigm, space-continuous analog systems etc. For discrete-event modeling some of those features are: variable number of formal signals and user-defined signal attributes. Object orientedness of AleC++ is a very useful feature, since modeling is object-oriented problem exploiting extensively inheritance and encapsulation. AleC++ gives the opportunity of both the hardware and software signal processing can be described in the same environment, that is very useful from a design verification point of view.

Alecsis simulator accepts mixed-language descriptions, enabling one part of description to be written in AleC++ and exploiting its important advantages, while other part (libraries in VHDL or model cards in SPICE) may be given in standardized form. At the moment a VHDL-AMS compiler for Alecsis is being implemented.

Here several aspects of the development of this new software are covered such as: extension of the AleC++ language, connection to the non-linear DC simulation, small-signal model generation, equation formulation in the frequency domain, equation solution with sparsity considerations, performance computation (including: amplitude, phase, real and imaginary parts, logarithmic scales, amplifications, local sensitivity etc.)

Simulation examples will show the state of the development and will include passive and MOS circuits.

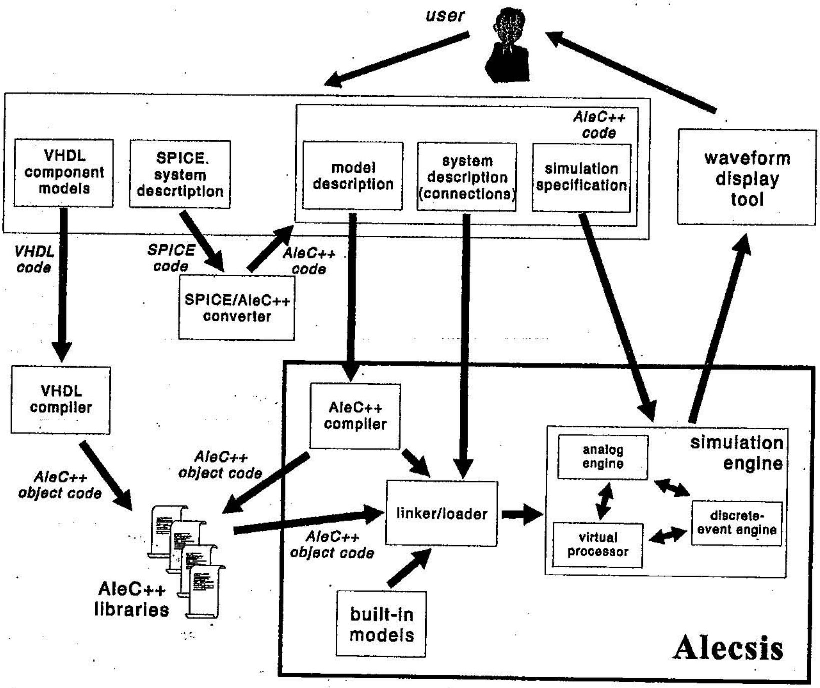

Figure 1. Alecsis simulator.

2. EXTENSION OF THE ALEC++ LANGUAGE

In order to implement AC domain simulation input language of ALECSIS simulator had to be extended. First a new command ac for AC analysis has been added having the parameters:

`fscale`	0 - linear 1 - dec 2 - oct
`fstart`	lowest frequency
`fstop`	highest frequency
`fnum`	number of points in which analysis occurs

E.g. `ac {fscale=2; fstart=10; fstop=100M; fnum=100;}`

The AC analysis works with complex numbers, and since AleC++ is an object oriented language, all the values have been added following properties: `amp` (amplitude), `phase` (phase), `real` (real part) and `imag` (imaginary part). E.g. `output->phase` gives the phase of the `output`.

New types of independent generators have been added as well. Those are vac - voltage and cac - current generator. They accept following parameters: amp - amplitude and phase - phase of the signal. For example:

```
vac vin;
vin (node1, node2) {amp=1mV; phase=0rad;}
```

In order to implement extensions of AleC++ language the parser (parser is generated using yacc parser generator and lex lexical analyzer) and the compiler [3] had to be modified accordingly.

3. Equations Formulation and Solving

For AC analysis it would be possible to formulate a matrix with complex elements, in which case not many changes would be necessary in matrix filling. Since ALECSIS simulator is written in ANSI C that has no complex type a different approach has been implemented. In order to enable real arithmetics to be implemented the system of nodal equations

$$\mathbf{Ax} = \mathbf{z} \tag{1}$$

is usually transformed into:

$$\begin{bmatrix} \mathbf{A}_r & -\mathbf{A}_i \\ \mathbf{A}_i & \mathbf{A}_r \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{z}_r \\ \mathbf{z}_i \end{bmatrix}, \tag{2}$$

where $\mathbf{A} = \mathbf{A}_r + i \cdot \mathbf{A}_i$, $\mathbf{x} = \mathbf{x}_r + i \cdot \mathbf{x}_i$ and $\mathbf{z} = \mathbf{z}_r + i \cdot \mathbf{z}_i$. Here **A** is the (modified) nodal admittance matrix, **x** is the vector of the unknown node voltages and branch currents, while **z** is the vector of excitations [4]. All the matrices in (2) are of type double.

It should be noted that obtained matrix has twice the size and is sparser compared to the matrix for transient analysis.

4. CONNECTION TO THE NON-LINEAR DC SIMULATION

Semiconductor devices have strictly nonlinear nature. However, in some circuit situations, the characteristics of the device must be represented only in a restricted range of currents and voltages. In particular, for small variations around operating point, the nonlinear characteristics of the device can be linearized by means of the expansion in Taylor series truncated after the first order terms. Obtained model is called small-signal model.

Parameters used in small-signal model are calculated during DC analysis that must precede the AC analysis. Those parameters for all nonlinear devices in a circuit must be stored and used in equation formulation for small-signal models [5, 6, 7, 8].

5. SIMULATION RESULTS

As a simulation example for passive circuits, a ten stage crystal band-pass filter shown in figure 2 [9] has been simulated. Obtained amplitude characteristic is shown in figure 3.

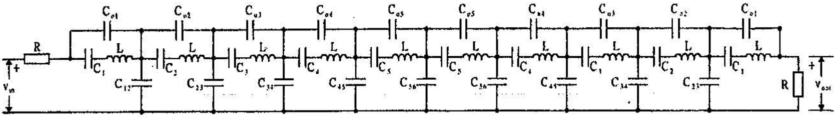

Figure 2. Ten-stage crystal band-pass filter.

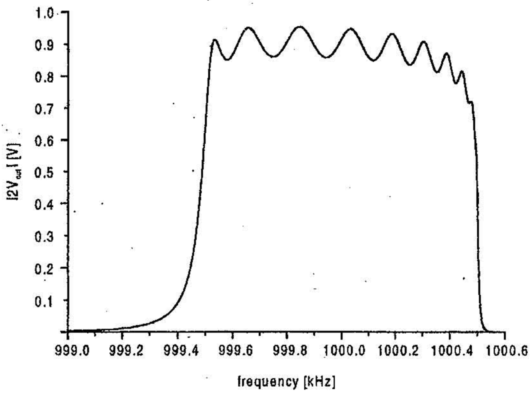

Figure 3. Amplitude characteristic.

As an example for simulation of active circuits, a simple one-stage MOS amplifier has been simulated.

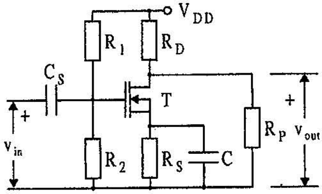

Figure 4. Simple MOS amplifier

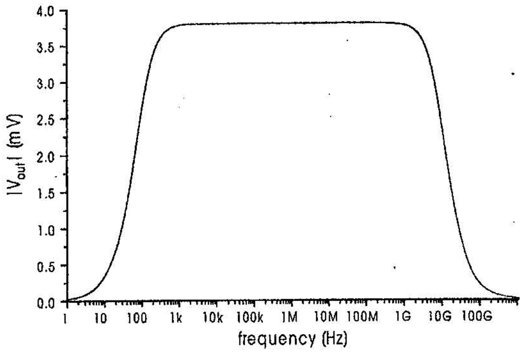

Figure 5. Amplitude characteristic.

Obtained results are in good accordance with SPICE simulations.

6. Conclusion

Basic considerations of adding AC simulation to ALECSIS simulator has been described in this paper. Given examples show the current state of development.

Small signal models for other semiconductor devices are being implemented, as well as linearization of needed parameters during DC analysis. Also, implementation of techniques for exploiting matrix sparsity in order to increase simulation speed is being considered [10]. Sensitivity analysis is also to be implemented.

7. References

[1] -----, "Alecsis 2.3, the simulator for circuits and systems. User's manual", Laboratory for Electronic Design Automation, Faculty of Electronic Engineering, University of Niš, Yugoslavia, LEDA – 1/1998.

[2] Litovski V., Maksimović D., and Mrčarica Z., "Mixed-signal modeling with AleC++: Specific features of the HDL", Simulation Practice and Theory 8, 2001, pp. 433-449.

[3] Aho A. V., Sethi R., and Ullman J. D., "Compilers - Principles, Techniques, and Tools", Addison-Wesley, 1986.

[4] Litovski V., and Zwolinski, M., "VLSI Circuit Simulation and Optimization", Chapman and Hall, London, 1997.

[5] Antognetti P., Massobrio G., editors, "Semiconductor Device Modeling with SPICE", McGraw-Hill, 1988.

[6] Getreu I. E., "Modeling the Bipolar Transistor", North-Holland, 1976.

[7] Divekar D. A., "FET Modeling for Circuit Simulation", Kluwer Academic Publisher, 1988.

[8] -----, "SPICE2 Version 3F.5 - Source Code", University of California, Berkeley, 1993.

[9] Humpherys D. S., "The Analysis, Design, and Synthesis of Electrical Filters", Prentice-Hall, 1970.

[10] Kundert K. S., "Sparse Matrix Techniques", in: Ruehli A. E., editor, "Circuit Analysis, Simulation and Design", North-Holland, 1986.