APPLICATIONS OF HARDWARE DESCRIPTION LANGUAGES FOR FREQUENCY DOMAIN MODELING AND ANALYSIS OF ANALOG CIRCUITS

Ivan Dimitrov Panayotov

Department of Electronics, Technical University of Sofia, 8, Kliment Ohridski St., 1000 Sofia, Bulgaria, phone: +359-2-965-31-15, e-mail: idp@ecad.tu-sofia.bg

In this paper the way of using universal Analog and Mixed Signal (AMS) Hardware Description Languages (HDLs) to model and simulate frequency dependant elements such as filters is discussed. Analog description languages such as VHDL-AMS and Verilog-AMS gain popularity as instruments for mixed signals modeling and simulations. They make it easy to model systems on high behavior level of description, to perform mixed-domain simulations and to design using modern system on chip approach. As standard languages in most their use, they allow more complex simulations to be made with universal HDL-AMS simulators and to use different abstraction levels for creating more or less detailed descriptions.

More professional CAD software tools such as CADENCE, SYNOPSYS, Mentor Graphics and others, provide HDL-AMS support in their design flow.

Keywords: VHDL-AMS, Verilog-A/AMS, Analog modeling, Simulations, Behavior descriptions, CADENCE, Mentor Graphics.

1. INTRODUCTION

The complexity of contemporary electronic designs urge for more advanced modeling and simulation tools and methods to support the design flow. Almost every electronic circuit contains analog, digital and mixed signal modules and the total functionality at system level can be verified by complex simulations for every domain and module. In such situations of significant help might be the use of Hardware Description Languages (HDL) for Analog and Mixed Signals (AMS). The most popular of them are VHDL-AMS and Verilog-A/AMS. They provide unified standards for creating pure digital and mixed signal simulations with new behavioral and structural language constructs and new simulation mechanisms as they appear to be supersets to the previous pure digital HDL languages VHDL and Verilog. Analog extensions give a possibility to describe and model complex electronic circuits and systems on a chip using the same language for digital, analog and mixed signal parts. Also approved design principals applicable to digital SoC may be transferred to analog circuits using descriptions of IP modules with proven functionality. All this aims at the high efforts to assure fast and faultless design flow.

Applying high level behavior models to analog RF circuits provides a mechanism for early design verification on system level.

Development of analog and mixed signal hardware description languages also encourages the efforts to provide the ability to automatically generate circuit from a description, the ability present for the digital design for many years. Even still far from achieving digital design synthesis performance, some tools exist that allow

automatic circuit generation based on text description in field of filter synthesis. They are based on using standard cells – operational amplifiers in particular, which can be parameterized to achieve required functionality.

In this paper the way of using VHDL-AMS and Verilog-A – the most popular and widely implemented in the CAD tools, mixed signal HDLs for modeling analog filters and amplifiers in frequency domain is discussed. The basic principals of using mixed signal HDLs are represented and simulations are carried out with professional CAD tools such as CADENCE and Mentor Graphics.

2. DESIGN PRINCIPALS

Hardware description languages VHDL-AMS and Verilog-A add the ability to model continuous aspects of systems behavior by means of system of Differential and Algebraic Equations (DAEs), with time as the independent variable. They have the form represented in (1):

(1) $F(x, dx/dt, t) = 0$

Here F is the vector of expressions, x is the vector of unknowns, dx/dt – the vector of derivate of the unknowns with respect to time, and t – time. For this in both languages new constructs for description and simulation rules are defined.

VHDL-AMS provides a general mechanism for describing DAEs by using "quantity" elements – bound to ports or free, new operator to assign values to quantities – "= =", this gives the designers the ability to write equations mathematically, with support of both implicit and explicit formulations.

In Verilog-A/AMS language the notation based on so called "probe-source network" is provided. The difference is made between the free and potential/flow quantities, and different assignment operators are used – "=" and "<+".

In both languages the concepts of a continuous-time, or analog, simulation kernel is present and both support the specification of initial conditions and the definition of piecewise-defined behavior (different regions of operations, between which the model can dynamically switch during simulation). To ensure stable and predictable model behavior mechanisms to handle discontinuities are included.

Both languages VHDL-AMS and Verilog-A/AMS provide functions for describing analog behavior in frequency domain. These functions represent the Laplace transformation equations. They provide a way of creating models for frequency domain analysis of analog blocks such as filters and amplifiers. The use of these functions for both languages is conducted by the rules for using analog operators. Most popular commercial CAD tools provide support for these functions.

In the VHDL-AMS language the Laplace transformation is represented by the operator:

quantity_name'LTF (num , den)

It represents the transfer function with nominator and denominator coefficient values given as real vectors – "num" and "den", respectively. The "quantity_name" denotes a quantity over which the transformation is performed. It could be any valid quantity type.

In Verilog-A/AMS there are several functions to represent the Laplace transformation depending on equation type:

laplace_zp(expr, zeta, rho) - to implement the zero-pole form Laplace transformation; zeta is a fixed-sized vector of M pairs of real numbers (each pair represents a zero); the first number in the pair is the real part of the zero, and the second is the imaginary part; rho is a fixed-sized vector of N real pairs, one for each pole.

laplace_zd(expr, zeta, d) - to implement the zero-denominator form Laplace transformation; zeta is a fixed-sized vector of M pairs of real numbers (each pair represents a zero); the first number in the pair is the real part of the zero, and the second is the imaginary part; d is a fixed-sized vector of N real numbers that contains the coefficients of the denominator.

laplace_np(expr, n, rho) - to implement the numerator-pole form Laplace transformation; n is a fixed-sized vector of M real numbers that contains the coefficients of the numerator; rho is a fixed-sized vector of N pairs of real numbers (each pair represents a pole); the first number in the pair is the real part of the pole, and the second is the imaginary part.

laplace_nd(expr, n, d) - to implement the numerator-denominator form of the Laplace transformation; n is a fixed-sized vector of M real numbers that contains the coefficients of the numerator, and d is a fixed-sized vector of N real numbers that contains the coefficients of the denominator. The values should depend only on parameters, and not on variables.

In this paper "laplace_nd" function is used to represent numerator-denominator form of the transformation, as it is the closest version to the VHDL-AMS "'LTF(nom,den)" function.

The models are based on the definition of the vectors of the coefficients based on the filter transfer function

(2) $$H(s) = V_{OUT}(s)/ V_{IN}(s)$$

The coefficients for the nominator and denominator are derived and implemented in VHDL-AMS and Verilog-A descriptions, as parameters of the transform functions.

3. RESULTS

The following figures represent the obtained simulation results from implementing High-pass filter and Operational amplifier in VHDL-AMS and Verilog-A, respectively in Mentor Graphics - System Vision and CADENCE - Analog Environment software tools. Respective codes are given in tables.

In both environments the models are implemented as model descriptions and symbol representations are created. They are used in a schematic to realize the test circuits.

3.1. High-pass filter

The model represents the transfer function to model the high-pass filter. The descriptions for VHDL-AMS and Verilog-A are given in the table 1.

The obtained simulation results – AC analysis are given on figures 1 and 2.

Table 1 VHDL-AMS and Verilog-A codes for high-pass filter.

VHDL-AMS description	Verilog-A description
entity LowPassFilter is generic (Fp : real := 1.0e6 K : real := 1.0 Q : real := 1.0); port (terminal input : electrical; terminal grnd : electrical; terminal output : electrical); end entity LowPassFilter; architecture ideal of LowPassFilter is quantity vin across input to grnd; quantity vout across iout through output to grnd; constant wp : real := math_2_pi*Fp; constant num : real_vector := (0.0,0.0,1.0); constant den : real_vector := (wp*wp, wp/Q, 1.0); begin vout == K * vin'ltf(num, den); end architecture ideal ;	`include "constants.h" `include "discipline.h" module high_pass_filter(x, y, g); inout x, y, g; electrical x, y, g; parameter real K = 1.0; parameter real Qf = 1.0; parameter real fp = 1.0e6; parameter real wp = fp*6.28318530717958647652; analog begin V(y,g) <+ K*laplace_nd(V(x, g), {0.0,0.0, 1.0}, {wp*wp, wp/Qf, 1.0}); end endmodule

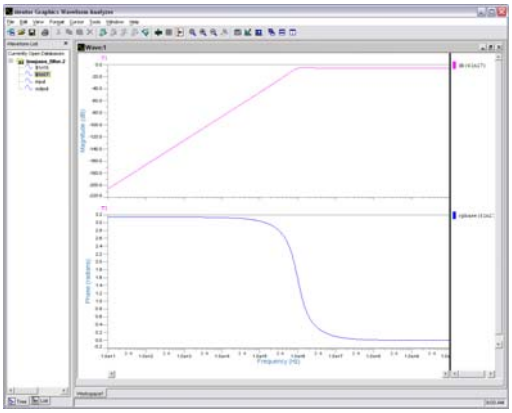

Fig. 1 High-pass filter VHDL-AMS model in MG SystemVision.

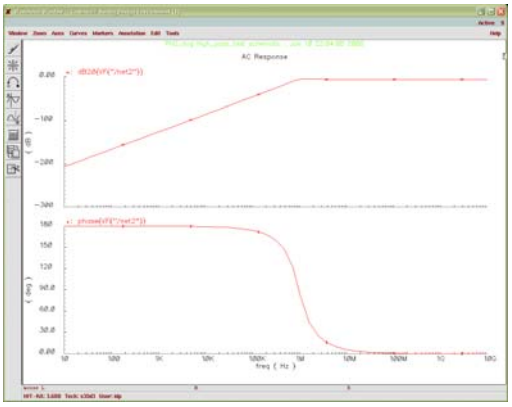

Fig. 2 High-pass filter Verilog-A model in CADENCE.

3.2. Amplifier

The simplest description of an amplifier just represent the gain over the input signal to produce the output value – equation (3).

(3) Vout = Vin*gain

This is does not represent any behavior over frequency. Such behavior could be modeled using Laplace transformation in the description of the amplifier. VHDL-AMS and Verilog-A codes are given in table 2. The obtained simulation results – AC analysis are given on figures 3 and 4.

Table 2 VHDL-AMS and Verilog-A codes for amplifier.

VHDL-AMS description	Verilog-A description
entity OpAmp_3p is generic (rin : resistance := 1.0e6; rout : resistance := 100.0 avol : real := 100.0e3; f_0dB : real := 1.0e6); port (terminal in_pos, in_neg, output : electrical); end entity OpAmp_3p; architecture basic of OpAmp_3p is constant f_3db : real := f_0db / avol; constant w_3dB : real := math_2_pi*f_3dB; constant num : real_vector := (0 => avol); constant den : real_vector := (1.0, 1.0/w_3dB); quantity v_in across i_in through in_pos to in_neg; quantity v_out across i_out through output; begin i_in == v_in / rin; v_out == v_in'ltf(num, den) + i_out*rout; end architecture basic;	`include "constants.h" `include "discipline.h" module opamp_freq_dom (in_pos, in_neg, out_put); inout in_pos, in_neg, out_put; electrical in_pos, in_neg, out_put; parameter real rin = 1.0e6; parameter real rout = 100.0; parameter real oplgain = 100.0e3; parameter real f0dB = 1.0e6; parameter real f3dB = f0dB/oplgain; parameter real w3dB = f3dB*6.28318530717958647652; analog begin I(in_pos, in_neg) <+ V(in_pos, in_neg)/rin; V(out_put) <+ laplace_nd(V(in_pos, in_neg), {oplgain}, {1.0, 1.0/w3dB}) + I(out_put)*rout; end endmodule

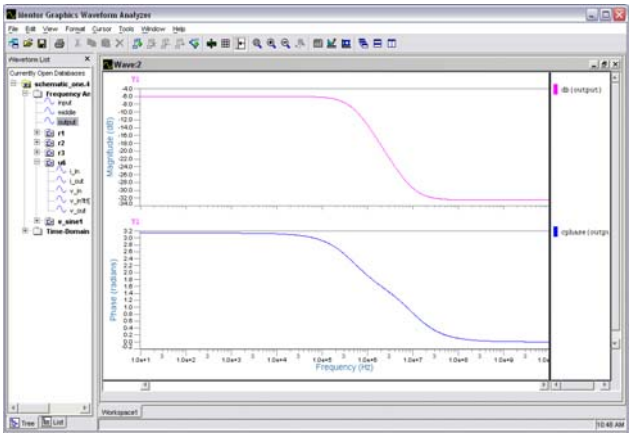

Fig. 4 Amplifier VHDL-AMS model in MG SystemVision.

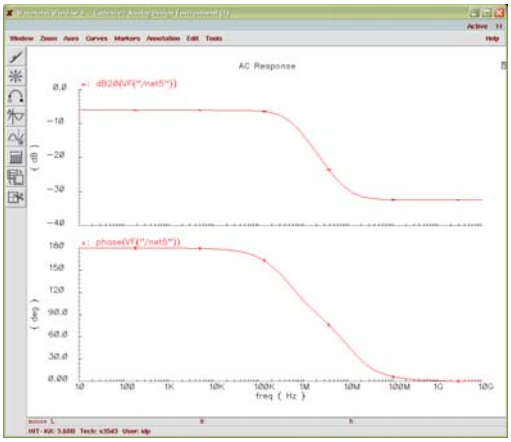

Fig. 3 Amplifier filter Verilog-A model in CADENCE.

In these descriptions not only input and output voltages are defined, as for the filter, but also input and output currents. The frequency response is defined by the selecting the appropriate values for the coefficients of the Laplace transformation.

4. CONCLUSIONS

In this work the way of creating behavior HDL models for analog models of frequency dependent elements was presented. The results were proven for the two most popular analog and mixed signals description languages – VHDL-AMS and Verilog-A, and for two of the leading simulator tools providers – Mentor Graphics and CADENCE. The obtained results proved the applicability of the languages and

software tool to create descriptions and provide simulations on high level for system on chip design applications.

The corresponding VHDL-AMS and Verilog-A models provide same results on two different platforms – Mentor Graphics System Vision and CADENCE Analog Environment. The defining of characteristics is done by means of parameters in both languages. This provides a way for easy model change. Using such kind of descriptions in complex systems design allows for fast check of different design decisions within minimum simulation time.

The descriptions are inserted in the CADENCE design environment as models with corresponding symbols creation, representing standard simulation cells, available for reuse.

5. ACKNOWLEDGMENTS

The research described in this paper was carried out within the framework of contract 08045NI-3.

6. REFERENCES

[1] Christen, E., Bakalar, K., "VHDL-AMS - a hardware description language for analog and mixed-signal applications", IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 46, (10), Oct.1999, pp. 1263 –1272.

[2] Panayotov, I., Ph. Philipov, Behavior description of elements of the structure of analog array, ET 2007 Proceedings, book 4, pg.59-64, ISBN 1313-1842

[3] Affirma™ Verilog®-A Language Reference, CADENCE Design Systems.

[4] VHDL-AMS Language Reference Manual

[5] Verilog-A/AMS Language Reference Manual