

## **Towards Analog-VHDL : Some of the Problems for Mixed Simulation**

H. EL TAHAWY  
ANACAD  
51 chemin du vieux chêne  
38240 MEYLAN ZIRST  
FRANCE  
Tel : +33 76 90 78 62

D. ROUQUIER  
CNET / CNS  
BP98  
38243 MEYLAN Cedex  
FRANCE  
Tel : +33 76 76 41 31

D. RODRIGUEZ  
CNET / ANACAD

### **ABSTRACT**

Computer simulation has become an indispensable tool in the design of VLSI circuits. Traditionally, a number of levels ranging from high-level behavioral to low-level electrical description have been successfully used to model and simulate digital circuits. In the past, modeling and simulation of analog circuits were restricted to the electrical or transistor level description, although this level is CPU-time consuming. More recently, however, an emerging need for the simultaneous representation of electronic circuits (analog or digital) at more than one level of description has spurred an intense research effort in multi-level modeling and simulation. Mixed-mode simulation i.e. simulation which simultaneously combines analog and digital descriptions, has been particularly prominent in this fast evolution.

For the digital domain, VHDL is a powerful and now widely accepted standard (for both modeling and simulation). There is also an urgent need for analog designers to have a standardized modeling and simulation language. Despite the major differences in concept between analog and digital modeling, a unified modeling capability seems to be possible especially at the behavioral level of abstraction. In this paper we discuss the different concepts of analog and digital modeling, and the requirements needed to describe both in a unified mixed-signal modeling language. The results of this work are preliminary ideas about future analog feature in VHDL (AHDL for Analog Hardware Description Language or VHDL-A).

## **Towards Analog-VHDL : Some of the Problems for Mixed Simulation**

H. EL TAHAWY  
ANACAD  
51 chemin du vieux chêne  
38240 MEYLAN ZIRST  
FRANCE  
Tel : +33 76 90 78 62

D. ROUQUIER  
CNET / CNS  
BP98  
38243 MEYLAN Cedex  
FRANCE  
Tel : +33 76 76 41 31

D. RODRIGUEZ  
CNET / ANACAD

### **-1- Introduction**

The rapidly escalating costs of designing and manufacturing VLSI circuits make design verification prior to production imperative. Despite the research progress of formal verification, simulation seems to remain the most practical design verification method. The objective of the simulation is to verify that the circuit under design will perform correctly in its operating environment.

A number of simulation levels, i.e. models and their corresponding algorithms, ranging from high-level behavioral to low-level electrical, have been successfully used for digital circuits. Unfortunately, for analog circuits, there is a lack of a description and simulation language at higher levels. So, a need arises for simultaneously simulating the circuit at more than one level [17]. Multi-level simulation techniques [2] can satisfy this need. Mixed-mode simulation [2] is also being developed, to enable the designers to trade simulation accuracy for simulation speed for the non-critical circuit components. VHDL was defined and is used mainly for digital systems. The FAS [3] modeling language written for the ELDO [4] circuit simulator is a modeling facility of analog and digital circuits at the behavioral level.

Based on the experience using VHDL and FAS, this paper discusses the possibility to model and simulates analog and mixed (analog/digital) circuits at the behavioral level while maintaining their critical characteristics. Section 2 introduces the basic concepts in simulation and modeling of analog circuits with respect to digital ones. Section 3 is devoted to the different modeling aspects of mixed (analog/digital) circuits. In section 4 we present what could be the future analog VHDL. Then section 5 rapidly presents some examples of mixed descriptions.

### **-2- Analog versus Digital Modeling and Simulation**

The analog simulation modeling at the circuit level is a system of coupled nonlinear algebraic differential equations. This model is usually specified in terms of interconnected lumped two-terminal linear or nonlinear elements such as resistances, independent and dependent sources and capacitances. The simulation model is constructed by combining the constitutive relations of these primitives along with the constraints of interconnection (Kirchoff's Current and Voltage Laws : KCL and KVL). The time domain behavior of the circuit is simulated by determining the voltage and current at each component's terminals at each point in time.

The digital simulation modeling is a system of de-coupled discrete algebraic difference equations specified in terms of interconnections of lower level logic or higher level functional primitives. These modules are characterized by unidirectionality and a de-coupling of their temporal and logical components. The time domain response is viewed as a succession of events.

Analog modeling levels are not standard and may be defined as behavioral, functional (or macro modeling) or primitives (transistor). Digital modeling levels are standard and defined as behavioral, functional (or RTL), logical, switch or transistor. Examples of existing analog modeling and simulation tools include SPICE [5], ASTAP [6], ELDO [4] and SABER [7]. Examples of existing digital modeling and simulation tools include LSIM [8], VERILOG and HILO. VHDL [13] has a place of its own, because it is a standardized language supported by several different simulators.

### **-3- Mixed (Analog / Digital) Modeling and Simulation**

It's difficult to blend analog and digital simulators because they're fundamentally different types of tools. A circuit simulator generates simulated waveforms and applies them to transistor-level models. The simulator then analyzes such analog characteristics as voltage over time, frequency, noise and temperature. For this purpose, the simulator solves differential equations for circuit nodes through a matrix inversion and makes successive approximations until it converges to a solution.

Digital logic simulators assign '0's or '1's to the selected nodes of a software model and then observe the results. The simulator looks at the functionality of the circuit and at timing relationships between circuit elements. Instead of continuous waveforms, logic simulators are concerned with discrete logic states and with events (changes in state at the inputs or outputs of the circuit).

Mixed-mode or mixed signal simulations have recently attracted a lot of attention for the following reasons :

- for enabling designers to make appropriate cost/accuracy tradeoffs within the design process.
- for allowing the simulation of hybrid(analog/digital) circuits (telecommunication, video, audio).
- for allowing the simulation of the whole circuit at different levels of abstraction.

Some basic issues of the mixed-mode simulation are :

#### 1) Implementation approach :

There are three approaches : the global-scheduler approach (which takes separate analog and digital simulators and establishes a communication mechanism between them), the environment-extension approach (which extends either a digital simulator to handle behavioral analog models, or an analog simulator to handle logic-level digital models) and the unified-environment approach (which means a complete integration of two algorithms in one simulation environment to handle both analog and digital circuitry).

#### 2) Synchronization method :

The analog simulator uses dynamic time step control, whereas the digital simulator works with events. The time at which these events must be considered does not necessarily coincide with the time when the analog simulator calculates a new time step. There are several methods for synchronization.

In the leap-frog approach of synchronization, the analog simulator is permitted to run as far ahead in time as it needs to. When a digital event occurs that requires analog processing, the simulator reevaluates its solution and either continues forward in time or adjusts backwards.

In the lock-step approach a unified time step control method is used. The digital simulator generates "Wake up alarm" each time it needs to evaluate its events (this time

is a multiple of the analog time step), so the analog simulator adjusts its time step with respect to the future "Wake up alarm" and does not need backtracking.

### 3) Signal representation and signal mapping

Digital simulators deal with binary values as 0,1 and even X for single bit. At the other extreme, analog simulators use 64-bit double precision words to represent voltage values. A mixed mode simulation must be able to manage these different signals and map them from one representation to the other without any significant loss in accuracy.

There are two kinds of translation : the analog to digital and the digital to analog, and two ways : explicit and implicit. In the explicit way, interface components can appear in the description file, these components do not have physical existence within the circuit. In such a solution the user can choose between several solutions and maybe can write his own model. In the other way these translations can be implicit, the analog and digital kernels know their interface connections and the translation is hidden to the user within the simulation.

### 4) Partitioning :

Circuit partitioning is a key factor in obtaining accuracy from the mixed-mode simulation. The main question is to determine which part of the circuit is critical and must be simulated at the transistor level and which part is non critical and may be simulated at a higher level of abstraction. The process to perform this task automatically seems formidable and this is still an "open" research area.

### 5) Unified modeling language :

Most analog simulators use a SPICE-like language (structural language), while the digital simulators can use behavioral and structural languages. The advantage to use a unified language will affect on the user interface, library construction, simulation debugging and schematic interface. This point is also "open" for research and will be discussed in the next section.

Some examples of mixed-mode simulators which already exist are : SPLICE [2], DIANA [9] , MOTIS [10], LSIM [8], Cascade [2], and FIDELDO [11].

## **-4- Overview of the possible future Analogue VHDL**

### NOTE :

This presentation reflects the views of the authors, and has not yet been endorsed by the IEEE working groups. Nevertheless it reflects most of the requirements received to date. Of course, examples are in a provisional syntax.

#### a Analog structural description

Interconnecting analog blocks or digital ones is not so different, thus part of the present day VHDL syntax for the structural description can be kept for analog structure. This is the case for the **block**, **component**, **configuration**, **generic** and **generate** statements.

However analog connections are not equivalent to digital ones. Namely a digital signal has only one value and when you connect it to one or several ports, all values become equal (except the problem of the driving and effective values). On the other hand, an analog wire is composed of two values, one **across** and one **through** (the current and the voltage for the electrical world). When you connect several analog terminals together, the through values (voltage) are equal (KVL) and the sum of the across values (current) is equal to zero (KCL).

To represent these analog-specific semantics, new syntactic constructs have been proposed [12]. The analog terminal is called **pin**, it is the equivalent of digital **ports**. The analog wire is called **node**, it is the equivalent of the digital **signal**. Connecting several **pins** to a **node** implies that KCL and KVL are verified.

#### b Analog behavioral description

Modeling the function of an analog subcircuit in a behavioral (not structural) way is necessary, at least for the basic components. It is also a key issue to reduce CPU time [17]. The **process** in VHDL-87 or VHDL-92 has a very precise simulation semantics, and is not convenient for analog parts. This aspect is linked to the resolution method used within the simulator, and two different methods have been proposed : modeling by equations, and modeling by procedural models.

In the first style, the behavior of an analog subcircuit is represented by an un-ordered set of differential equations indicating the relationship between voltages and currents at the subcircuit's terminals (or **pins**). This style allows simple and natural modeling of most frequently-used parts ; in some cases it is the only possible method, but it leads to a greater number of equations in the equation set.

In the second style, the behavior of an analog subcircuit is represented by a procedure-like piece of code, returning the branch currents as functions of the node voltages (and of their time derivatives). In some cases, also voltages can be computed as functions of currents.

In the first style, the equations given by the user are added to the set of equations to be solved by the simulator ; other equations come from applying KCL to each node. In the second style, the procedures given by the user are called by the simulator one or several times at each time-step (to find node voltages satisfying KCL). Both methods have been adopted, with different syntactic forms, by MAST[7] and FAS[3], two behavioral analog HDLs.

#### c Other features

An analog model can be controlled by the value of the current through a pin or the voltage difference between two pins, without interfering with this current or voltage. So to model this control a new interface clause was proposed with its corresponding **map** clause : the **parameter** clause [12]. Also new basic operators are needed, such as the time derivative (to express differential equations) and mathematical functions (e.g. sine, cosine ...). At last, an important feature could be the verification of the dimension of an analog expression[15]. This implies reviewing the current definition of physical units and of their management . In summary, each expression would have a unit which would be verified during compilation. This feature would avoid error in the description of a model whose voltage or current expressions are difficult. (should TIME be included in this reviewing or will there be a time for simulation and a time for expressions ?).

#### d Example of a behavioral analog model : an OpAmp

The main points of this example are the declaration of the pins, the use of the new physical types, the time derivative operator, the time step and the analysis mode (DC or AC) of the simulator. The formula used to compute the VI voltage is obtained by approximating the time derivative with the explicit Euler scheme in the following expression :

$$\frac{1}{2 \cdot \pi \cdot FC} \frac{dVI}{dt} + VI = EPSI \text{ then } \frac{1}{2 \cdot \pi \cdot FC} \frac{(VI - VI'old)}{TSTEP} + VI = EPSI$$

Obviously the equivalent internal scheme of this Op-Amp is an RC circuit given by the cut frequency. Note that the GND pin is in fact the ground, to ensure that the sum of the currents flowing through the pins is equal to zero.

```

entity OP_AMP is
  generic
    (VSATP, VSATM, OFFSET : physic unit is VOLT;
     SL : physic unit is VOLT/SECOND;
     RIN, ROUT : physic unit is OHM;
     GAIN : REAL;
     FC : physic unit is HERTZ);
  pin
    (EP, EN, S, GND : ELECTRICAL)
end AMPLI_OP

architecture BEHAVIORAL of OP_AMP is
begin
  analog_process
    variable VI, EPSI : physic unit is VOLT;
  begin
    EPSI := GAIN * (EP.V - EM.V);
    if (MODE=DC) then VI:= EPSI;
    else VI:=(TSTEP*2.0*PI*FC+1.0)*(EPSI+VI'old/(2.0*PI*FC*TSTEP));
    if (d_dt(VI)>SL) then VI:= VI'old + SL*TSTEP;
    elsif (d_dt(VI)<SL) then VI:= VI'old - SL*TSTEP;
    end if;
  end if;
  if (VI>VSATP) then VI:=-VSATP;
  elsif (VI<VSATM) then VI:=-VSATM;
  end if;
  EP.I:=(EP.V-EM.V)/RIN;   EM.I:=(EM.V-EP.V)/RIN;
  S.I:=(S.V-VI-OFFSET)/ROUT;   GND.I:= -(S.V-VI-OFFSET)/ROUT;
end analog_process;
end BEHAVIORAL;

```

### e equivalent structural model of the Op-Amp

```

architecture STRUCTURAL of OP_AMP is
  node N1, N2, N3: ELECTRICAL;
begin
  R1 : RESISTOR
    generic map (RIN)   pin map (EP, EN);
  E1 : CONTROLLED_VSOURCE
    generic map (GAIN)   parameter map (EP, EN)   pin map (N1, GND);
  R2 : RESISTOR
    generic map (1.0 OHM) pin map (N1, N2);
  C : CAPACITOR
    generic map (1/(2.0*PI*FC*(1.0 OHM)))   pin map (N1, GND);
  V2 : LIMIT_VSOURCE
    generic map (VSATP, VSATM) parameter map (N2, GND)   pin map (N3, GND);
  R3 : RESISTOR
    generic map (ROUT)   pin map (N3, S);
end STRUCTURAL;

```

This structural example is equivalent to the behavioral description of the OpAmp in the previous paragraph, except for the slew-rate. It can be seen that it is very similar to the description of digital structures, with new, specific, key-words.

### f behavioral description by equations

```

architecture EQUATION of RESISTOR is
begin
  relation
    P.V - N.V - R*P.I = 0.0 V ;
    P.I + N.I = 0.0 A ;
  end relation ;
end EQUATION;

```

This example shows how to model a resistor using the "equation" style ; the two equations are :

- first : OHM's law ;
- second : the two pin currents are equal (with opposite signs)

(NOTE : most probably, mixed simulators will use "hard-wired" models for resistors, capacitors, ...).

### -5- Mixed examples

A mixed model is a model which uses digital constructs and analog constructs (e.g. ports and pins and parameters, or signals and nodes, or processes and "analog-processes" and "relations"). The classic examples are the interface components which are used to translate the values between the two worlds.

```

entity A2D_INTERFACE is
  generic (   V01, V10 : physic unit is VOLT ;
             T01, T10 : TIME  );
  port      (D : out BIT);
  parameter (A : ELECTRICAL); -- infinite input impedance
end A2D_INTERFACE;

architecture A of A2D_INTERFACE is
begin
  process      begin      wait until (A.V > V01);   D<= '1' after T01;
                                wait until (A.V < V10);   D<= '0' after T10;
  end process;
end A;
-----
entity D2A_INTERFACE is
  generic (V1, V0 : physic unit is VOLT; TRISE, TFALL: physic unit is SECOND);
  port      (D : in BIT);
  pin       (A : ELECTRICAL)
end D2A_INTERFACE;

architecture A of D2A_INTERFACE is
begin
  analog_process
  begin
    if (D = '1') then      A.V:= SLOPE(A.V'old, V0, V1, TRISE);
    else                    A.V:= SLOPE(A.V'old, V1, V0, TFALL);
    end if;
    A.I:= 0 A;
  end analog_process;
end A;

```

Both models are very simple. The A2D uses a **parameter** to model infinite input impedance, and uses a digital **wait** on an analog value ; in this case a thresholding function should be provided, to avoid unnecessary calls to the digital kernel. The D2A uses two different slopes for the rising and falling edges, but they do not depend on the load on the node.

### -6- Conclusion

Multi-level and mixed-mode simulation are increasingly recognized as useful methods for the validation of VLSI circuits. For the description of digital parts, a standard (VHDL) has emerged and is now supported by several simulators. On the other hand, for the description and simulation of analog parts, the existing languages (apart from the standard, SPICE, which is purely structural) are numerous and proprietary. For the description of mixed parts, the situation is also difficult. Therefore the main needs are :

- a standardized functional (not only structural) language for the description of analog circuits ;

- a standard way of describing mixed analog-digital circuits. This would allow users to build and exchange "universal" models (not linked to one proprietary language and simulator) and simulator vendors to compete in an open way. Also model vendors would benefit from such a standard. The existing analog models (mostly in Spice) would not be lost, because purely structural descriptions can easily be translated from one syntax to another.

It has been shown in this article that some aspects of digital and analog modeling, at the structural and functional levels, are similar (and that others are very specific). Work is currently going on, within IEEE working groups, to specify an analog behavioral language (sub-par 1076.1) . It will certainly be difficult to define these features in a safe and efficient way, but given the existing functional languages (for analog circuits) and the existing mixed-mode simulators, it should be possible.

## References

- [1] D. Borrione et Al, "Overview of the CASCADE multi-level HDL and its mixed-mode simulation mechanisms", CHDL'85.
- [2] A.R.Newton, "Techniques for the simulation of LSI circuits", IEEE Tran. Circuits Syst.Vol CAS-26, pp. 741-749, Sept. 1979.
- [3] "the ELDO-FAS modeling language, reference manual", see ANACAD
- [4] B.Hennion et al, "A new algorithm for third generation circuit simulators : the One Step Relaxation method", Proc 22nd DAC, 1985.
- [5] L. W. Nagel, "SPICE2, a computer program to simulate semiconductor circuits", University of Berkeley, ERL Memo N° ERL M520.
- [6] W.T. Weeks et al. ; "Algorithms for ASTAP : a network analysis program"; IEEE Trans. Circuit theory , Vol CT20 pp 628, Nov 73.
- [7] I. Getreu ; "Analog modeling language spans all system design levels"; Electronic Design, pp 95-104, May 87.
- [8] P. Odryna ; "A unified mixed-mode digital/analog simulation environment"; ISCAS'88.
- [9] P.H.Reynaert et al, "DIANA : A mixed\_mode simulator simulator with a hardware description language for hierarchical design of VLSI", proc IEEE Int. Conf. Circuits & Computers, Port Chester, Oct 1980, pp. 356
- [10] B. R. Chawla and Al., "MOTIS - An MOS Timing Simulator", IEEE Transactions on Circuits and Systems, vol. CAS-22, PP.301 December 1975.
- [11] H.El Tahawy et al, "A new implementation technique for the simulation of mixed (Digital-Analog) VLSI circuits", IEEE Conf on Computer Aided Design ICCAD'87.
- [12] C. Le Faou, J. Mermet ; Requirements E12, E13 and E14 for the rebalotting of VHDL-92
- [13] D. Borrione et Al ; Requirements E63 and E64 for the rebalotting of VHDL-92
- [14] D. Rouquier ; Requirements E16 for the rebalotting of VHDL-92
- [15] D. Rodriguez, D. Rouquier ; Requirements E62 for the rebalotting of VHDL-92
- [16] M. Brown ; Requirements A001 to A007 for the rebalotting of VHDL-92
- [17] G. Gielen et Al, "Analog behavioral models for simulation and synthesis of mixed signal systems", proc. EDAC'92 pp 464-468