Proceedings of the Workshop „Advances in Analog Circuit Design“, Villach, Austria, 26.-28. April, 1995, 10/1 - 10/22

MIXED-MODE DESIGN: EXPERIENCES WITH MULTI-LEVEL MACROMODELING

Peter Schwarz, Karsten Einwich, Joachim Haase, Rico Prescher

Fraunhofer Institute for Integrated Circuits
Department EAS Dresden / Germany
D - 01069 Dresden, Zeunerstrasse 38

Abstract

For simulation in mixed-mode design appropriate macromodels have to be developed. A successful approach is the combination of analog and digital, time-continuous and time-discrete, behavioral and structural models to „mixed-signal macromodels“. Submodels on different levels of abstraction are combined, too (multi-level approach). Due to the complexity of such models, an automatic generation is not possible. Instead, a heuristic approach is proposed and illustrated by some examples from industrial design tasks. In addition, the application of the forthcoming mixed-signal hardware description language VHDL-A is discussed.

1. INTRODUCTION

Macromodeling has proved to be a very effective way for the acceleration of simulation and the best way for system simulation. For building up cell model libraries, for the construction of system models of very large or complex integrated circuits, and for getting insight into circuit behavior, macromodels are very useful. Macromodels may be defined as models, which

- represent an approximation to the pin behavior of the original circuit,
- have not to represent the internal circuit behavior,
- can be constructed as structural models (equivalent circuits), behavioral models, or as a combination of both.

Macromodels of this type support the bottom-up design process - they are abstractions of transistor and/or gate level circuits. In the top-down approach the original specifications or behavioral descriptions of subsystems may be regarded as initial macromodels, which have to be refined in the design progress.

The first, „classical“ macromodels in the analog circuit domain were equivalent circuit models of operational amplifiers [1, 2]. Until today, this is the most widely used class of macromodels. In the digital domain, several types of models may be interpreted as macromodels: instruction set descriptions or high-level register transfer descriptions, truth tables, finite state machine descriptions, or VHDL behavioral descriptions.

In mixed-signal circuit design combinations of all these model classes have to be used. Consequently, it is worth looking for a general strategy to generate mixed-signal macromodels which use analog and digital signals and combine different modeling levels (multi-level macromodels). This corresponds to the growing use of mixed-mode simulators [3]-[7]. Universal macromodels should be suited for analog, digital, and mixed-signal simulators (after some slight simulator-dependent modifications, possibly). In this way the flexibility necessary for system simulation can be guaranteed.

2. MODELING STRATEGY

Due to the complexity and the functional differences between the circuits being investigated, the macromodeling procedure is basically a manual and heuristic one and, at present, far from being automated. In the analog domain, transient behavior, sometimes also DC behavior and AC behavior of linear subsystems (jω- and s-domain, poles/zeros) have to be modeled. In the digital domain, combinational and sequential behavior, delays, signal strengths, and different signal types (bits, words,...) have to be considered. The modeling strategy is roughly summarized in Figure 2-1. Depending on the accuracy and speed requirements and the levels of description offered by the simulator some iteration loops may be necessary. As mentioned above, this modeling approach is mainly carried out „by hand“ with numerous simulation runs and trial-and-error improvements. However, software tools like SimPilot or other modeling/simulation environments [8] and languages (e.g. CLANG [9]) may support the model optimization and parameter determination process.

Special attention should be paid to the analog-to-digital and digital-to-analog conversion of signals. Instead of applying very complicated behavioral models or sophisticated equivalent circuits we successfully used small parts of the original transistor circuitry together with very simple AD- and DA-converter models, even in the case of rather difficult I^2L interfaces.

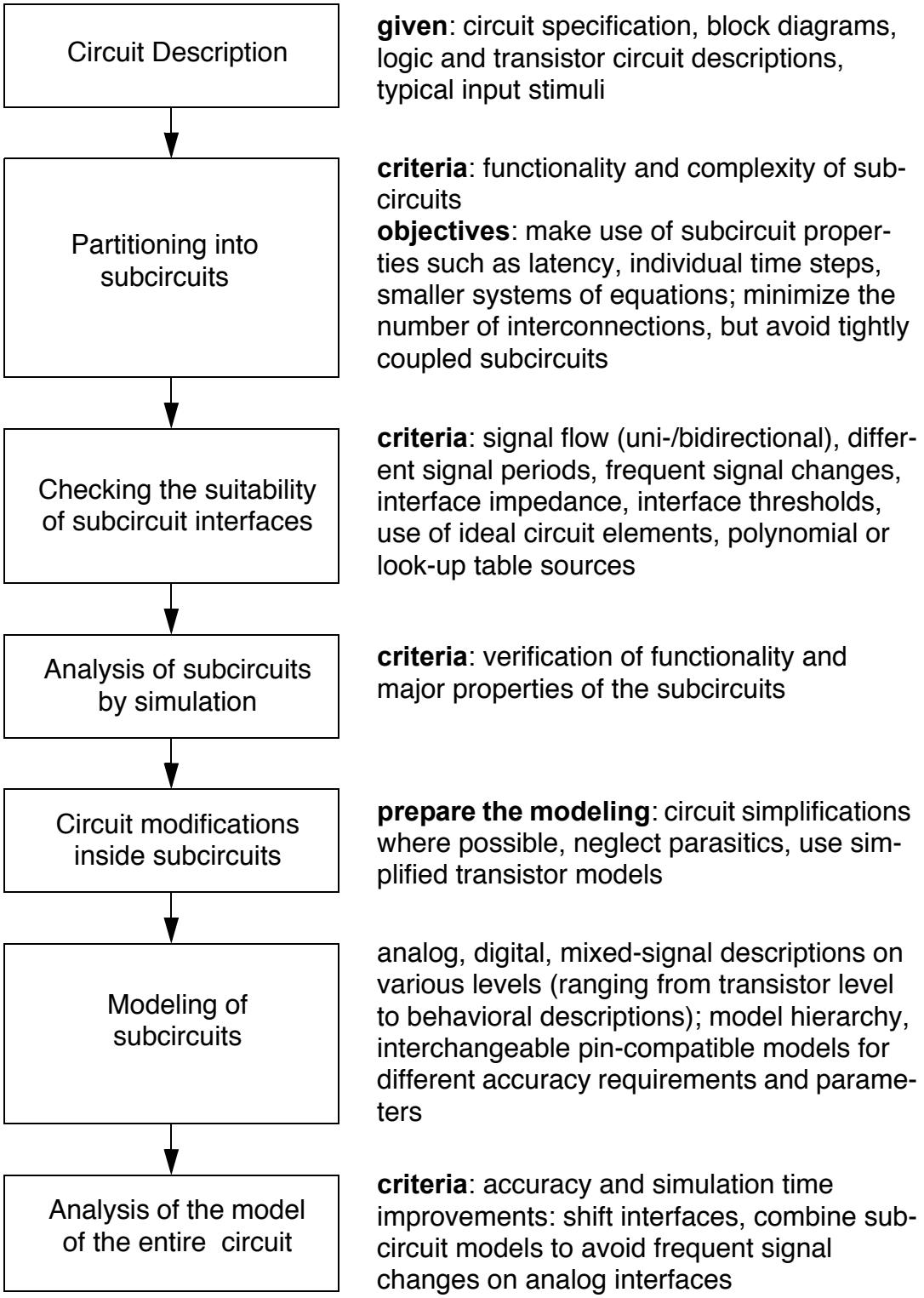


Figure 2-1: Steps in mixed-signal macromodeling

3. MODELING EXAMPLES

The modeling strategy shall be illustrated by some examples. Most of them are from design tasks of industrial partners (Siemens, TEMIC). They cover a wide range of modeling aspects and are therefore good examples of mixed-signal, multi-level macromodels. They shall demonstrate the main aspects of the heuristic modeling approach outlined in Figure 2-1:

a) Generation of macromodels of complex subcircuits for cell libraries. The cells and their interface signals have to be modeled accurately.

b) Generation of macromodels of the entire circuit (system model) for an overall simulation, which could be delivered to customers for simulations in their applications, too.

c) Combining analog and digital, linear and nonlinear, behavioral and structural submodels.

d) A nearly perfect match of transistor level circuits and behavioral macromodels can be achieved by using some boundary circuitry as part of the interfaces and AD / DA converters.

e) Different types of macromodels have to be considered:
   - models of digital circuits embedded in an analog environment,
   - models of analog circuits embedded in a digital environment,
   - models of analog-digital circuits in a mixed-signal environment.

The simulation runs were carried out with two standard circuit and system simulators which have mixed-mode capabilities: ELDO from ANACAD and SABER from Analogy with their native modeling languages. Also our experimental in-house simulator KOSIM [10] was applied.

3.1 Toggle IC

The Toggle IC (Figure 3.1-1a), developed by TEMIC, is used in automotive applications and performs signal recovering functions. Pulses generated by a switch connected to the input pin cause a relay at the output to alternately switch on and off. Set or reset pulses with a signal pulse width below approximately 80 milliseconds are suppressed (debounced) by means of digital logic.

An on-chip oscillator supplies the digital part with the necessary pulse frequency of 100 kilohertz. A frequency divider and digital logic that establishes the debouncing function form the digital part. All internal voltages as well as a power-on reset signal are delivered by the voltage reference block. Altogether, the Toggle IC comprises approximately 500 transistors.

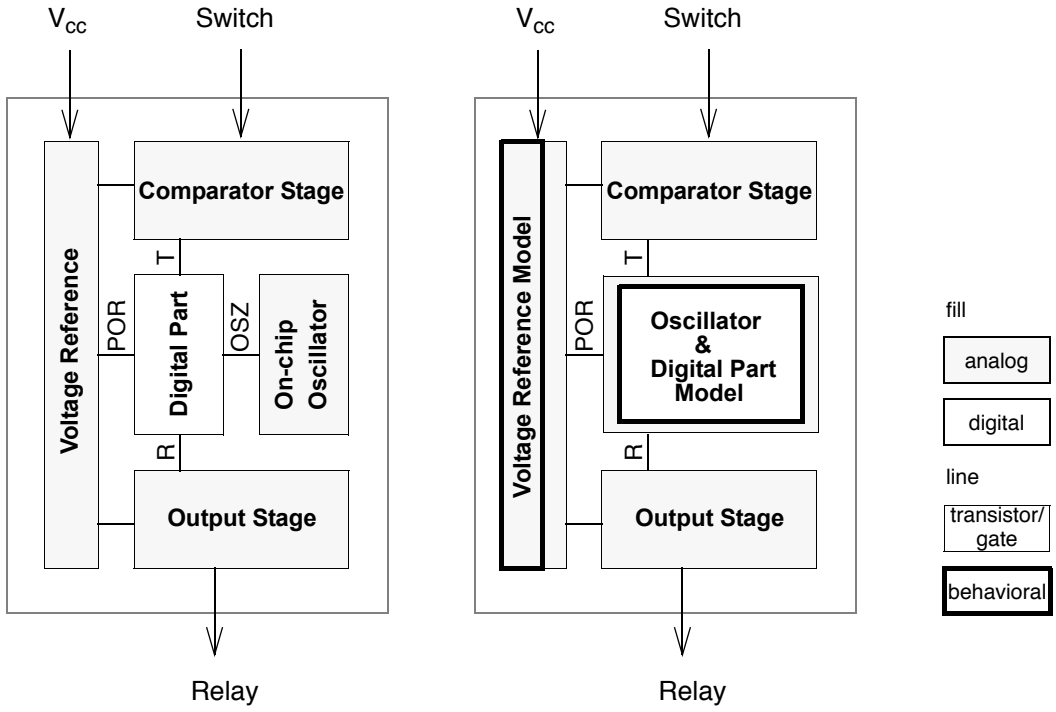


Figure 3.1-1: a) Block diagram and b) Macromodel of Toggle IC

The oscillator macromodel is formed by an analog behavioral model (including temperature dependencies) and by the output transistor as analog boundary. Within the voltage reference macromodel, one transistor at pin *POR* and the voltage controlled sources are combined with a digital behavioral model for generating the power-on reset signal. For the mixed-signal macromodel of the digital part (Figure 3.1-2), a digital behavioral model is embedded in a transistor environment via analog-to-digital and digital-to-analog converters.

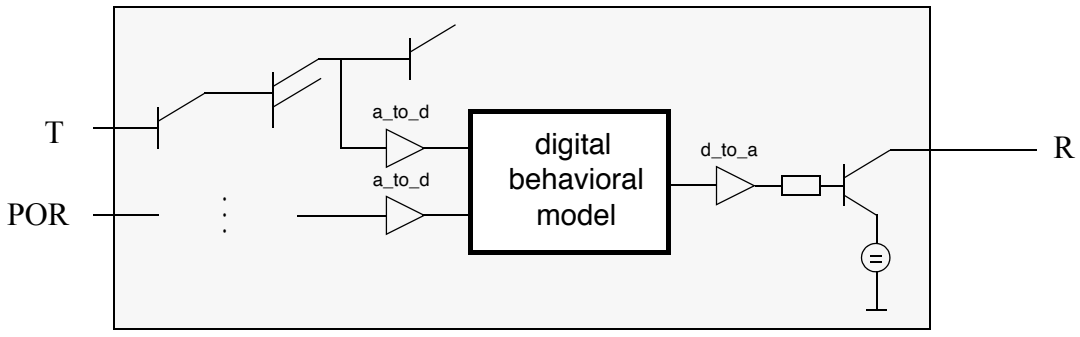


Figure 3.1-2: Digital part macromodel

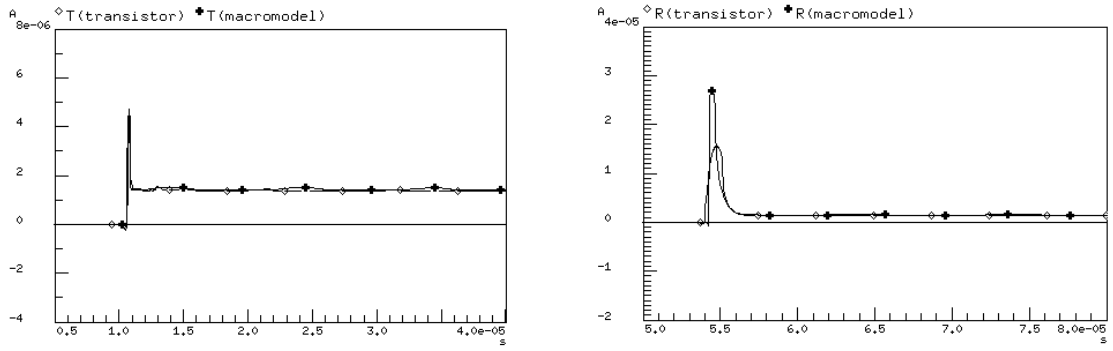


Figure 3.1-3 shows the very similar waveforms in the original transistor circuit and the macromodel.

Figure 3.1-3: Currents at the I^2L interface

Using a modified macromodel structure (see Figure 3.1-1b) that combines oscillator and digital part within one digital model, overall simulations can be achieved. Hence, there is no longer a „slow“ analog interface to be passed by the oscillator signal. Figure 3.1-4 confirms the correct operation of the entire circuit macromodel over a time interval of 0.5 seconds. Note that the first pulse in Figure 3.1-4 does not fulfil the minimum time requirement (data sheet values of 61 to 82 milliseconds) and therefore does not change the output state.

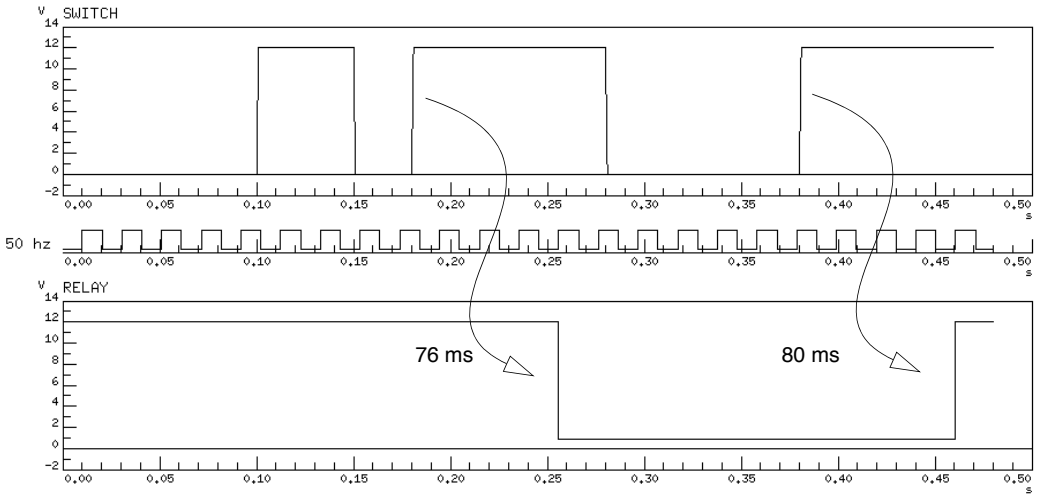


Figure 3.1-4 : Verification of the entire circuit macromodel

3.2 Gain Controlled Amplifier

The TEMIC Gain Controlled Amplifier GCA (Figure 3.2-1a) is used for amplifying and limiting a phase-modulated differential input signal within automatic

gain control loops. After passing the input and offset stage **IN** the signal to be amplified enters an eight-stage amplifier chain. A 3 bit digital signal controls the gain factor from 0 to 42 decibels in steps of 6 decibels. The output stage **OUT** is used for buffering and limiting the magnitude of the output signal. The GCA incorporates appr. 300 transistors. To derive the GCA macromodel (Figure 3.2-1b) from the given transistor level description, analog behavioral models were combined with parts of the transistor circuit.

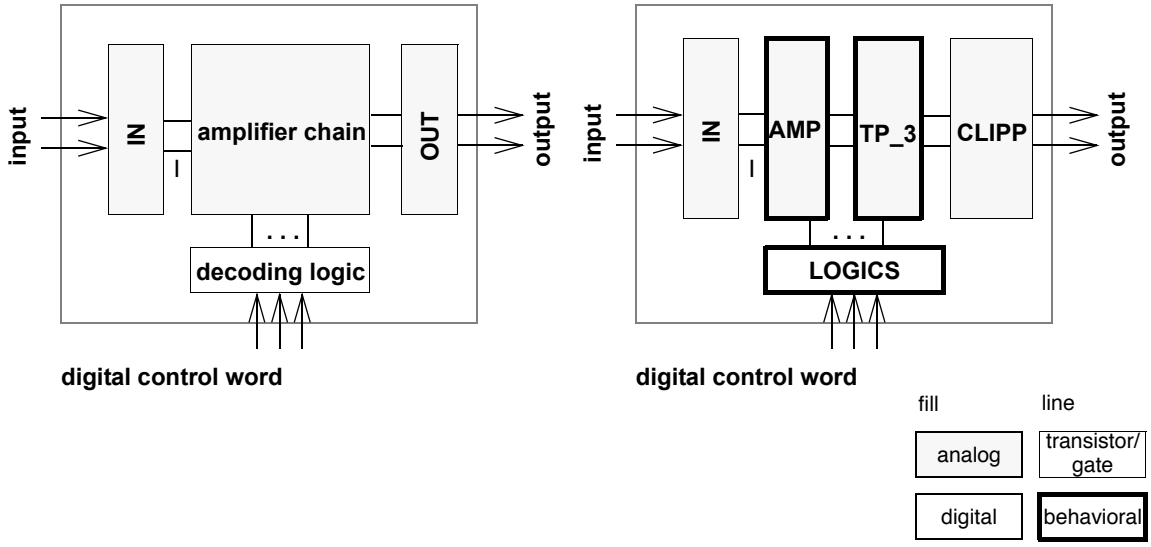


Figure 3.2-1 : a) Block diagram and b) Macromodel structure

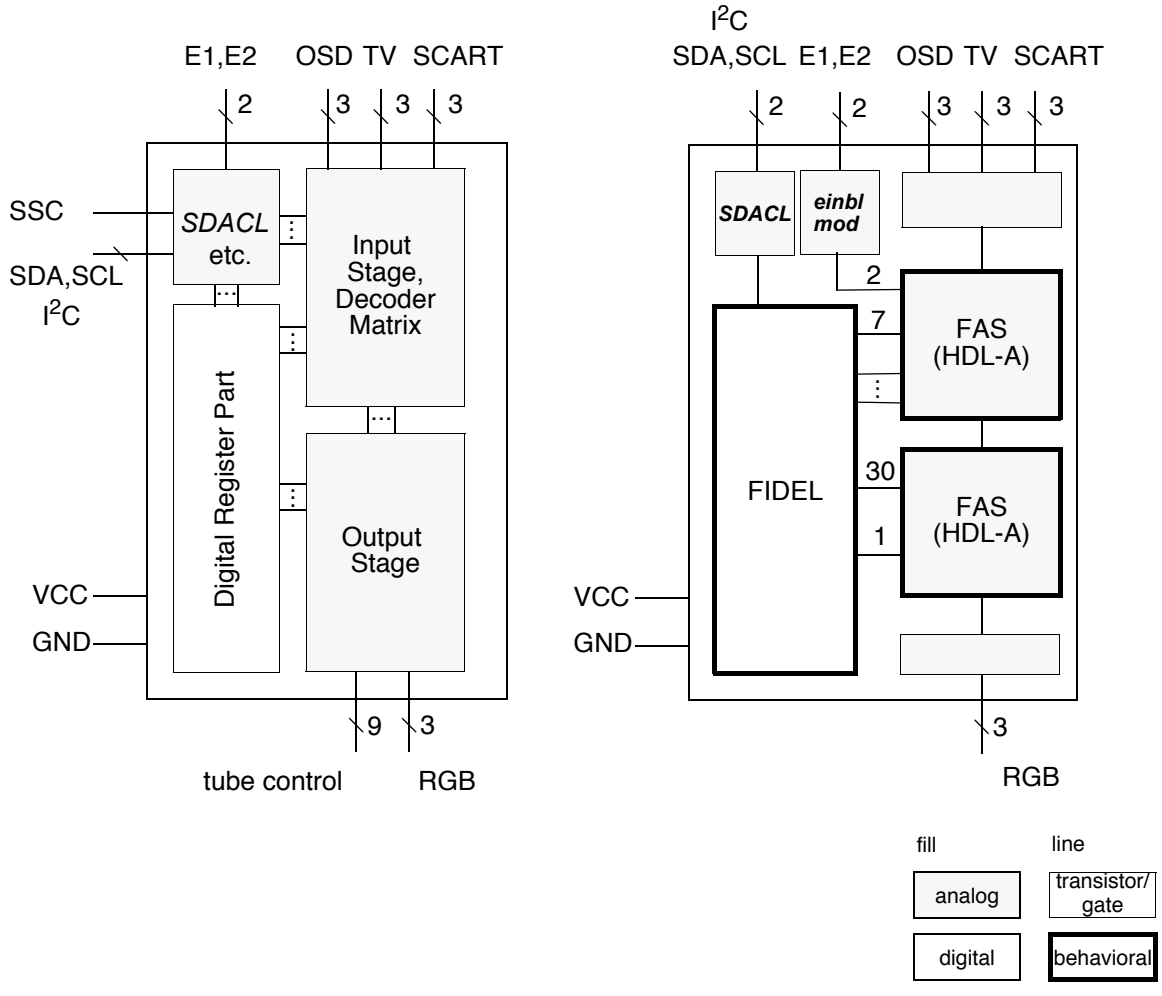
The input stage **IN** was taken from the transistor level circuit without any changes which seemed to be the simplest way to precisely model important input characteristics such as frequency-dependent impedances and zero locations of the transfer function. The behavioral model **AMP** constitutes a frequency-independent, temperature-dependent gain block representing the original eight-stage amplifier chain. Simulation showed that the chain's gain can be regarded as constant within the frequency range concerned. Thus, we developed a special model for the entire chain, without modeling every single stage.

The frequency response of the GCA turned out to be primarily determined by the frequency behavior of the output stage. A 3rd-order pole model with identical cutoff frequencies gives a suitable approximation and is contained in the behavioral analog model **TP_3**. The close correspondence between transistor level circuit and macromodel is demonstrated in Figure 3.2-2 (minimum and maximum gain). The block **CLIPP** consists of parts of the transistor level circuit (transistors, resistors, and capacitors) and controlled sources. To reproduce the clipping behavior of the output stage that is difficult to describe analytically, look-up tables are used in the controlled sources. The pairs of values for those tables were obtained by simulating the static transfer characteristic of the entire circuit

(Fig. 3.2-3). Figure 3.2-4 shows the dynamic clipping behavior for an input signal of 6.5 megahertz. The block **LOGICS** represents the decoding logic of the transistor circuit and incorporates its input transistors.

3.3. Videoprocessing Circuit

The TEMIC Videoprocessing Circuit (Figure 3.3-1a) is used in state-of-the-art television receivers. Only the major circuit features can be explained in this paper. These are the switching between three signal sources (*PAL/NTSC*, *SCART*, *On-Screen Display*) to the *R-G-B* output and processing of these signals. All control and adjustment signals are programmable via a serial two-wire bus (I^2C Bus). Data received across this bus is stored in a register array which holds the actual values of brightness, contrast, color contrast, gain etc. Again, our modeling activities aimed at two goals: generation of rather abstract models for the three main functional parts and, using these subcircuit models, generation of a macromodel for an overall simulation of the processor. With this model it is for the first time possible to investigate the videoprocessor‘s behavior in interaction with other parts of the television receiver.



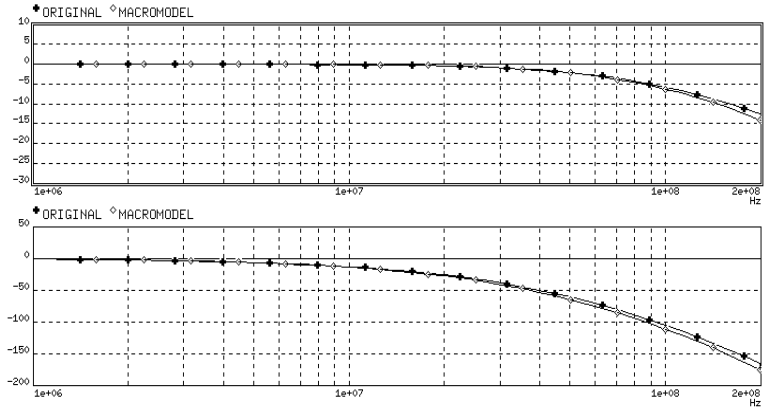


Figure 3.2-2a:
Frequency Response -
Unity Gain

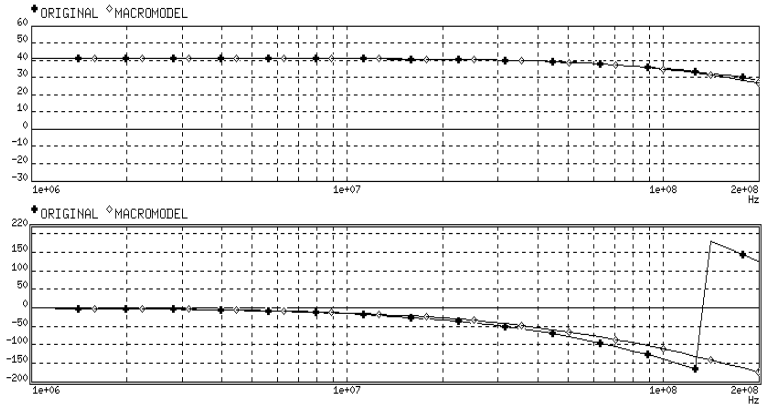


Figure 3.2-2b:
Frequency Response -
Maximum Gain

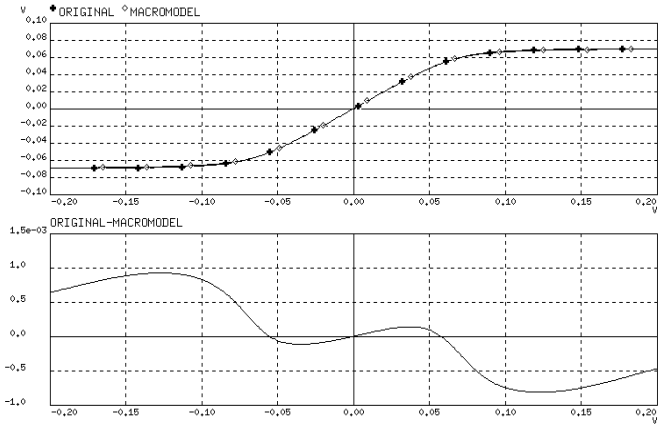


Figure 3.2-3:
Static Transfer Characteristic

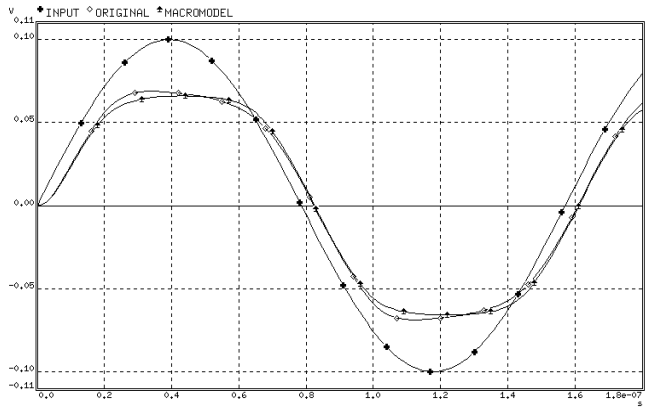


Figure 3.2-4:
Dynamic Clipping Behavior

Figure 3.3-1: a) Block diagram and b) Entire circuit macromodel

The **input part** (Figure 3.3-2) is used for switching between the input channels *TV*, *SCART* and *OSD* and allows the color contrast adjustment of the *TV* and *SCART* signals. For recovering the *R-G-B* signal from coded (PAL or NTSC) video signals a decoding matrix with adjustable coefficients is used. In the input stage macromodel, analog boundary circuits surround an analog behavioral model that reproduces the switching between input channels, matrix decoding, and the color contrast control. As a verification example, Figure 3.3-3 shows the color contrast control for a pulse applied to the *SCART* input. The output part is modeled accordingly.

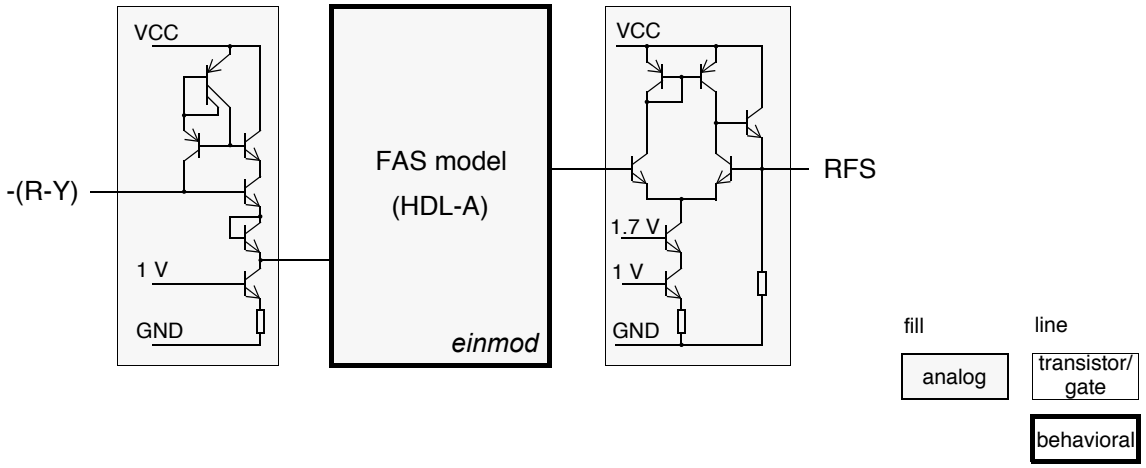


Figure 3.3-2: Macromodel of the input stage

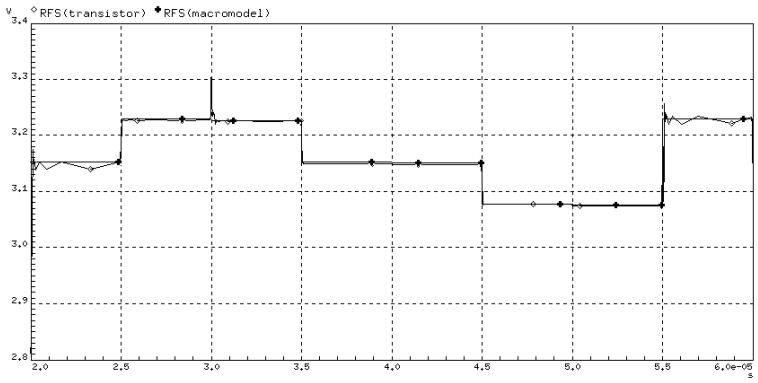


Figure 3.3-3: Verification of input stage macromodel

The main elements of the processor's **digital part** are a register array for storing data received via the serial I^2C bus (signals *SDA*, *SCL*) and a control part performing tube and power-on control. The digital behavioral model was obtained by analyzing the function of the digital part. To instantiate digital models in the SPICE-like netlists of ELDO, interface models are provided which map logical

(digital) values to analog ones and vice versa. By properly choosing the parameter values of these converters (i.e. threshold voltages, output voltages) the specific properties of the digital part interfaces (implemented in I^2L technology) to the input and output stage can be modeled precisely.

To form the macromodel of the entire circuit (Figure 3.3-1b), the mentioned subcircuit models have to be combined with each other and with smaller subcircuits remaining on transistor level. Various combinations of macromodels and transistor/gate level subcircuits can be found, depending on what part of the entire circuit requires most attention.

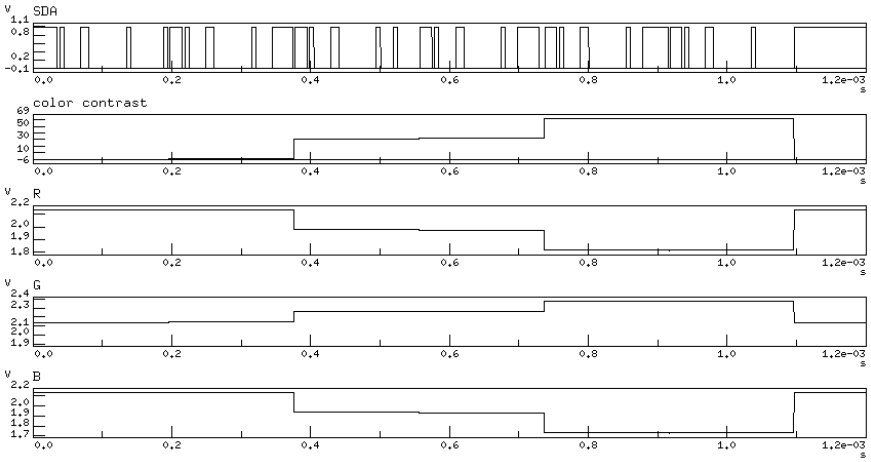


Figure 3.3-4: Color contrast adjustment

Figure 3.3-4 illustrates the color contrast adjustment controlled by the I^2C bus. Data input *SDA*, outputs *R*, *G*, *B* and the actual color contrast values are depicted. Due to frequent signal changes at the clock input *SCL* (clock of 6 microseconds period) the use of the transistor level block **SDACL** has considerable influence on simulation speed. If this block is omitted, i.e. the digital part inputs are stimulated directly by a digital clock source, transient simulations are accelerated up by a factor of 10.

3.4. Subscriber Line IC

In co-operation with Siemens EZM Villach behavioral models and a simulation environment for the building blocks **SLIC**/**SICOFI** [11, 12] (Figure 3.4-1) were developed.

ICs of this type are used in digital switching units on the analog subscriber line interface. They consist of various parts with quite different requirements on simulation techniques, namely:

- analog parts (transistor level netlists, filters on behavioral level)
- digital (control) parts (gate level netlists, behavioral descriptions)

- digital (filter) parts:
  although implemented digitally, their function is analog:
  transient and small-signal behavior (TR, AC) is of interest
  they are realized in hardware or DSP
  two different definitions of DC behavior are possible
  (digital initialization, equivalent analog dc value)

Simulating the SLIC includes the following problems:

- extremely different signal frequencies (see Figure 3.4-2):
  (sigma-delta-converter : megahertz, AC voice signal : kilohertz, DC control signals : hertz)
- strong feedback loops in the SLIC IC (convergence problems, very cpu-time intensive)
- tight coupling between analog and digital parts of the SLIC.

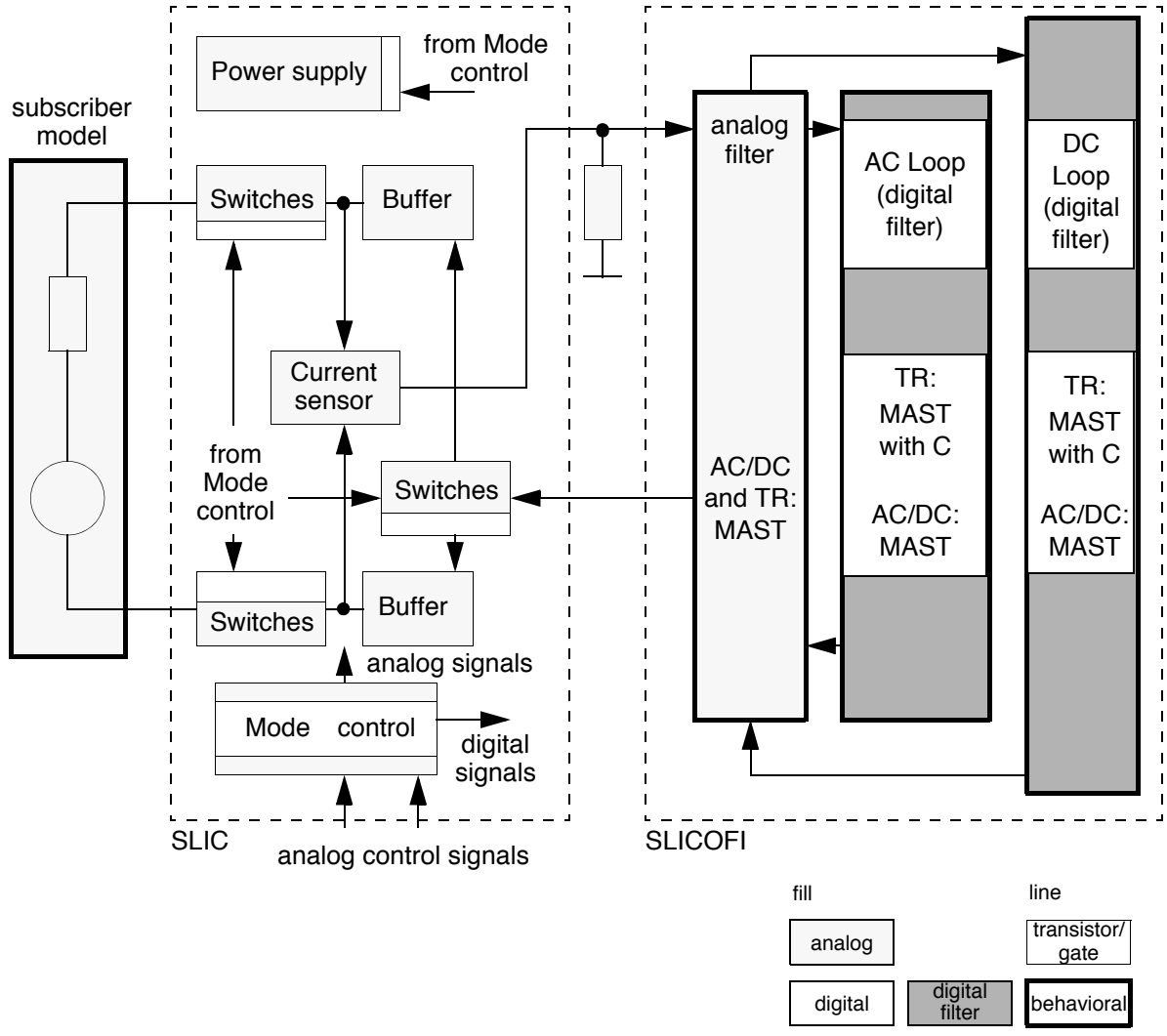


Figure 3.4-1: Functional block scheme

Input data for modeling the filter parts were derived from system descriptions in MathCad-, MatLab/Simulink format or from filter maps. For the SLIC a netlist was used. The models generated were implemented in SABER, using the native language MAST and procedures in C.

For the SLIC exchangeable models on three levels of abstraction were developed. These models are combinations of:

- MAST analog and digital behavioral descriptions,
- digital behavioral descriptions in MAST and C,
- netlists on transistor level.

Choosing the appropriate description was done by analysing the function in order to find a trade-off between accuracy and simulation run-time. For simulation control a set of script files for controlling different modes of the whole chip set was written. It was possible to completely automate the transformation of digital filters from system simulation (data stream) to TR-, AC- and DC- overall simulations in SABER.

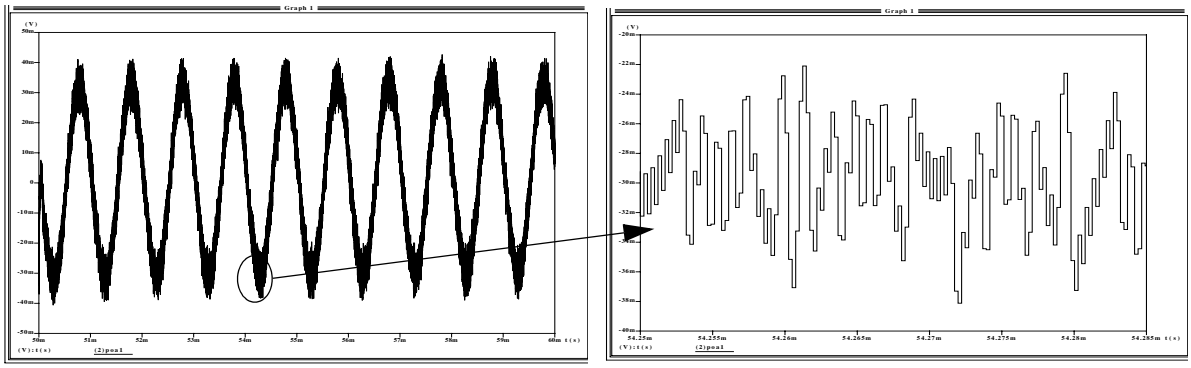


Figure 3.4-2: Signal at the ΣΔ converter output

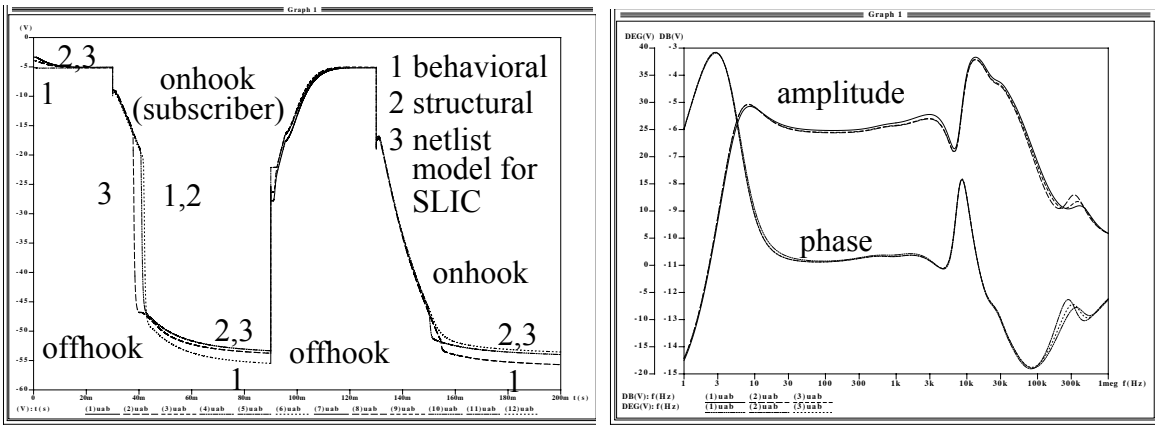


Figure 3.4-3: DC and AC characteristics with different SLIC models

3.5 PLA circuit

A MOS-PLA which may be embedded in an analog circuit environment is used for demonstrating the modeling strategy. Input and output inverters perform inversion and buffering, whereas two NOR matrices contain the logical function of the PLA circuit (Figure 3.5-1a). Thus, the two-level matrix part need not be described in detail at the transistor level, and a digital behavioral model can be applied.

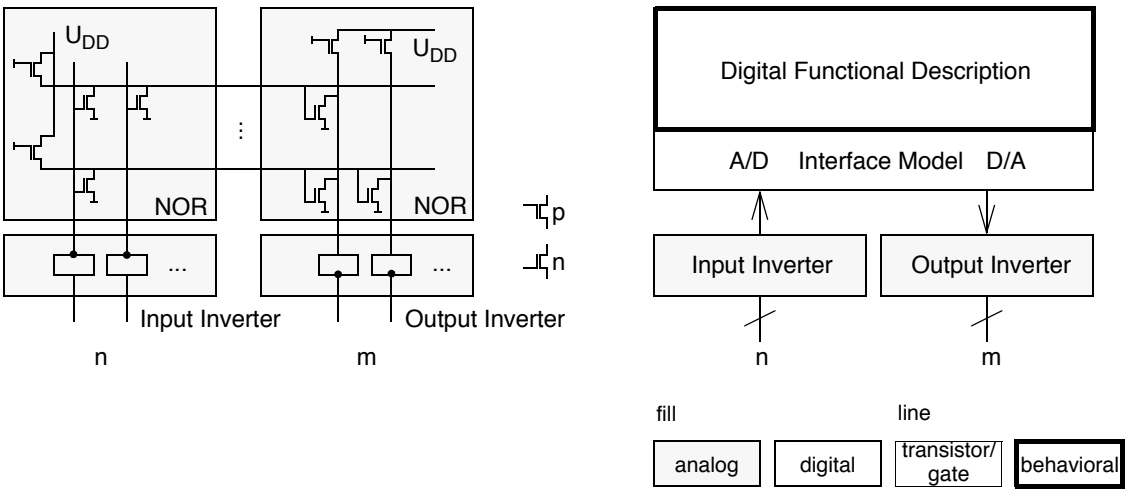


Figure 3.5-1 a) Digital circuit and b) Macromodel structure

Combining a digital behavioral model for the matrix part with inverters on transistor level results in a true mixed-mode model (Figure 3.5-1b), in which different circuit parts are represented on the most appropriate level. In this way a suitable trade-off between simulation speed and accuracy can be achieved in an analog environment, too. The mixed-mode PLA model was implemented in the experimental in-house simulator KOSIM as well as in ANACAD‘s ELDO. One possible method for automatically deriving behavioral models from a MOS transistor description is explained in [13].

For the extraction of signal delays, we tested SimPilot from ANACAD. Using the function *delaytime x y*, for instance, the signal delay between nodes *x* and *y* can easily be obtained. The values so determined are input to a C program that

calculates pin delays with a linear programming approach. The pin-to-pin delays were implemented in the behavioral model.

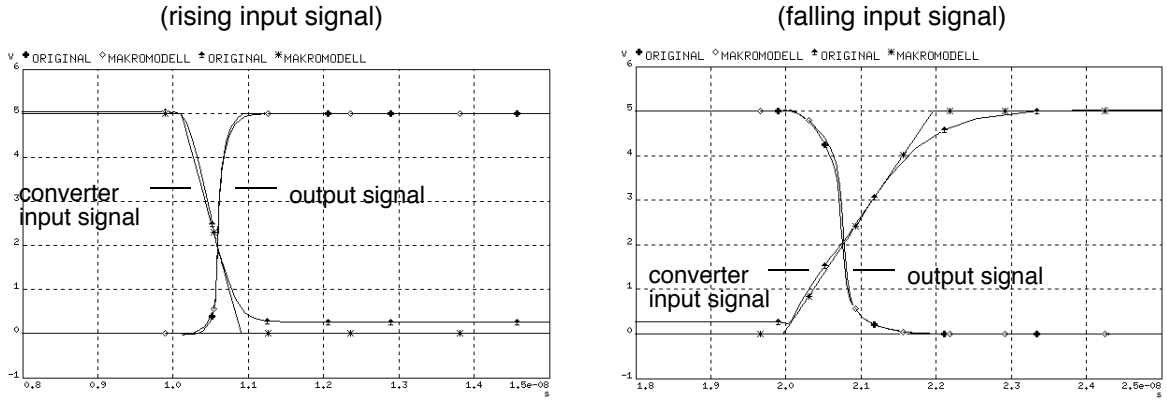


Figure 3.5-2 Verification of macromodel

In Figure 3.5-2 and 3.5-3 simulation results are shown both for analog and logic values. Although the signal waveforms before the output inverter circuit have some deviations from the original ones, both signals are nearly identical after the macromodel‘s inverter.

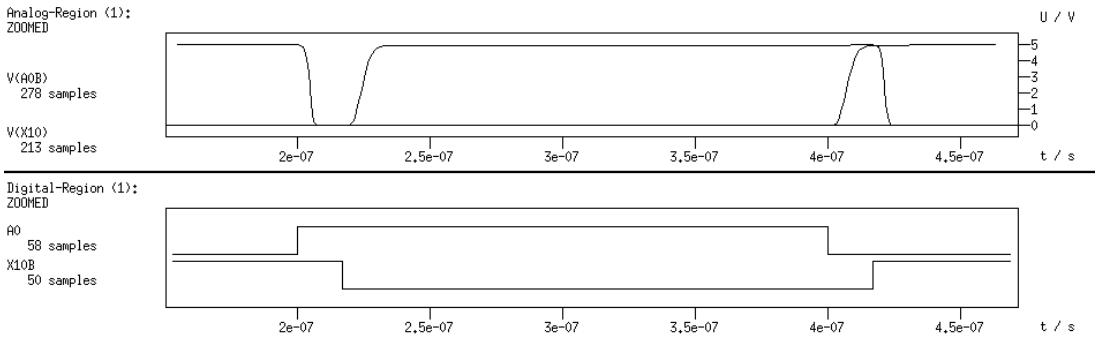


Figure 3.5-3 Simulation results

4. MIXED-SIGNAL HARDWARE DESCRIPTON LANGUAGES

4.1. The Needs for Standardized Descriptions

The mixed-signal macromodeling approach requires mixed-signal simulators with powerful modeling capabilities. In Table 4.1-1 an incomplete list of such simulators is given. Obviously, there is no chance to develop models which can be used by some or all of these simulators. (The only exception would be the usage of SPICE-compatible equivalent circuits, but this type of macromodels is not well-suited for mixed-signal design. Most of the SPICE-like simulators have

yet another restricted modeling capability: ideal voltage or current sources controlled by some other voltages or currents. But this feature is insufficient for general behavioral modeling.)

Thus, there is a need for a widely-accepted or standardized hardware description language with mixed-signal capabilities. One of the most promising attempts is the development of „Analog VHDL“ (VHDL-A) undertaken by the IEEE Standardization Committee 1076.1 [14, 15, 16]. VHDL-A is intended to be a superset of the digital VHDL, and will therefore be suitable for mixed-signal macromodeling. Until now there is no Language Reference Manual or another draft available which could be used for the evaluation of semantic or syntactic details. However, the main ideas are documented and available via anonymous ftp (nestor. epfl.ch, use read-only entry pub/vhdl/standards/ieee/1076.1). Unfortunately, the signal conversion between the analog and digital domain is not yet defined either. As a prototype, the new language „HDL-A“ of ANACAD´s simulator ELDO [5] may be used for modeling and simulation experiments, but other CAD vendors will release similar preliminary versions of VHDL-A-like languages.

Simulator	Vendor/ Developer	HDL (behavioral)	Type of Simulator
Apex	Intergraph	DIABOLO, C-like	analog
BONSIM	Bosch	C-Interface/PMI	analog PMI - Procedural Model Interface
ELDO	ANACAD	HDL-A, FAS, C Interface/CFAS	mixed-signal Coupling to Verilog, ...
KOSIM	FhG-EAS	C-Interface	mixed-signal, multi-level
PSpice	MicroSim	ABM	mixed-signal; Analog Behavioral Modeling with controlled sources
SABER	Analogy	MAST/C routines may be included	mixed-signal Coupling to Verilog, ...
SIMPLORER	SIMEC	Formulas, State graphs, C-Interface	mixed-language, multi-level: Petri nets, block diagrams, electrical circuits
SMASH	Dolphin	C-Interface	mixed-signal
Spectre	Cadence	Spectre HDL	analog
XSPICE	Georgia Inst Tech	C-Interface/Code Model Subsystem	analog extended version of SPICE 3

Table 4.1-1: Some simulators with behavioral modeling capabilities

4.2. A Modeling Approach

In addition to a mixed-signal modeling language a mathematically founded modeling approach is necessary [17, 18]. We propose a method which is based on the terminal behavior description of components/subsystems [19]. One of the main ideas of this approach is the separation of the terminals into six classes or types of terminals (see Figure 4.2-1):

A electrical pins with terminal voltages $\mathbf{v_a}$ and controlled (dependent) terminal currents $\mathbf{i_a}$:
modeling voltage-current relations in conventional nodal analysis,
B electrical pins with controlled (dependent) terminal voltages $\mathbf{v_b}$ and terminal currents $\mathbf{i_b}$: modeling the effects of ideal voltage sources,
C pins with input signals $\mathbf{a_{in}}$ (without current flow),
D pins with output signals $\mathbf{a_{out}}$ (without current flow):both for modeling block diagrams in control systems and signal flow diagrams,
E logical input ports with signals $\mathbf{d_{in}}$,
F logical output ports with signals $\mathbf{d_{out}}$.

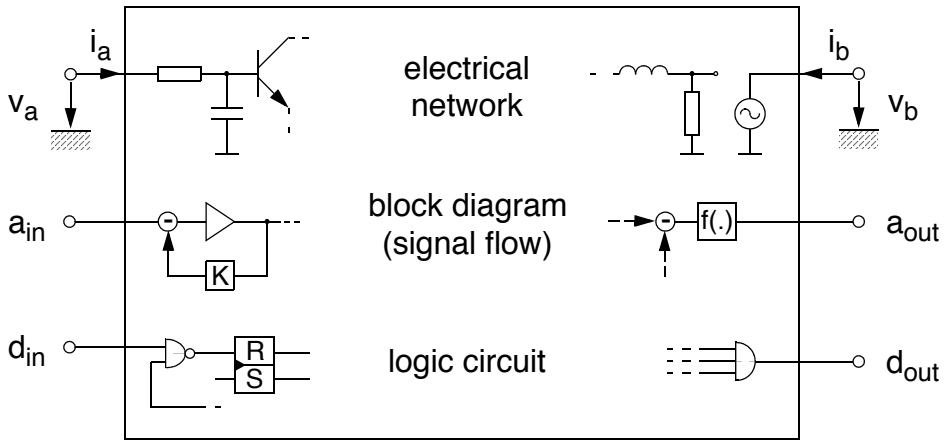


Figure 4.2-1 Definition of terminal signals

In general, all signals are time-dependent functions and are expressed as vectors. Pin classes C and D could be replaced by A and B, but the consideration of currents in block diagrams is not the normal modeling style. Only voltages are used in such systems as signals, and therefore Kirchhoff´s current conservation law is not applicable. Such „Non-Kirchhoff Nodes“ are also used in ANACAD´s HDL-A. The approach may be extended to bidirectional logical ports.

A mathematical description of such systems may have the following form:

The dot • denotes derivatives with respect to time d/dt, the vector **p** consists of system parameters, **s** is the vector of additional state variables. The introduction

$$\text{(1a)} \qquad \mathbf{i_a} = \mathbf{f_1}(\mathbf{v_a}, \dot{\mathbf{v}}_\mathbf{a}, \mathbf{i_b}, \dot{\mathbf{i}}_\mathbf{b}, \mathbf{a_{in}}, \dot{\mathbf{a}}_\mathbf{in}, \mathbf{s}, \dot{\mathbf{s}}, \mathbf{d_{in}}, \mathbf{p}, t)$$

$$\text{(1b)} \qquad \mathbf{v_b} = \mathbf{f_2}(\mathbf{v_a}, \dot{\mathbf{v}}_\mathbf{a}, \mathbf{i_b}, \dot{\mathbf{i}}_\mathbf{b}, \mathbf{a_{in}}, \dot{\mathbf{a}}_\mathbf{in}, \mathbf{s}, \dot{\mathbf{s}}, \mathbf{d_{in}}, \mathbf{p}, t)$$

$$\text{(1c)} \qquad \mathbf{a_{out}} = \mathbf{f_3}(\mathbf{v_a}, \dot{\mathbf{v}}_\mathbf{a}, \mathbf{i_b}, \dot{\mathbf{i}}_\mathbf{b}, \mathbf{a_{in}}, \dot{\mathbf{a}}_\mathbf{in}, \mathbf{s}, \dot{\mathbf{s}}, \mathbf{d_{in}}, \mathbf{p}, t)$$

$$\text{(1d)} \qquad \mathbf{0} = \mathbf{f_4}(\mathbf{v_a}, \dot{\mathbf{v}}_\mathbf{a}, \mathbf{i_b}, \dot{\mathbf{i}}_\mathbf{b}, \mathbf{a_{in}}, \dot{\mathbf{a}}_\mathbf{in}, \mathbf{s}, \dot{\mathbf{s}}, \mathbf{d_{in}}, \mathbf{p}, t)$$

$$\text{(1e)} \qquad \mathbf{d_{out}} = \mathbf{f_5}(\mathbf{v_a}, \dot{\mathbf{v}}_\mathbf{a}, \mathbf{i_b}, \dot{\mathbf{i}}_\mathbf{b}, \mathbf{a_{in}}, \dot{\mathbf{a}}_\mathbf{in}, \mathbf{s}, \dot{\mathbf{s}}, \mathbf{d_{in}}, \mathbf{p}, t)$$

of *additional* state variables is the key point in this „extended hybrid“ description in the electrical domain. It even offers the possibility to model very complicated nonlinear characteristics [20]. Calculation of the terminal signals requires solving of nonlinear differential-algebraic equations (DAE´s). To compute the state variables **s**, equation (1d) should be passed to the DAE-solver of the mixed-signal simulator. Therefore, the behavioral description language must have a mechanism for exchanging the values of **s** and $\mathbf{f_4}$ between model and simulator. For the digital part of the model, equation (1e) is usually represented by VHDL or VerilogHDL descriptions. However, in simple cases, boolean equations may be programmed directly. In equation (1e), the dependency from previous digital states and their updating is not expressed explicitly.

4.3 Realization of the Modeling Approach with HDL-A

Like the forthcoming VHDL-A, the language HDL-A has most of the properties to describe the terminal behavior with equation (1). Without going into detail, the structure of the HDL-A mixed-signal macromodel is :

```
ENTITY entity_name IS
    description of the component interface
    (pins, ports,parameters p)
END ENTITY entity_name;

ARCHITECTURE name OF entity_name IS
      declaration of
      additional states s, auxiliary variables, ...

    BEGIN RELATION
      PROCEDURAL
        calculation of
          dependent currents i_a ,
          dependent voltages v_b ,
          and output signals a_out
          in equations (1a, b, c)

      EQUATION (list of additional state variables s)
        formulation of equ. (1d) for bringing  f_4
        to zero with the simulator¥s DAE solver
    END RELATION

    PROCESS BEGIN
      calculation of d_out in a VHDL-like manner
    END PROCESS

END ARCHITECTURE name;
```

Example

The circuit *detector* (Figure 4.3-1) shall serve as a simple example. With a rising edge of the clock signal (port *clk*), the circuit detects whether the threshold *vs* is crossed by the filtered input voltage *inp.v*. Model parameters are input resistance, input capacitance, cutoff frequency of the filter, and the threshold value. The terminal voltage *inp.v* controls the input current *inp.i*. Similar to (1a) this current is computed in the PROCEDURAL block of the model (Figure 4.3-2). This block describes input resistance and capacitance connected in parallel. The output *lp_out* of the low-pass filter has to be introduced as additional state variable **s**. The low-pass with cutoff frequency *fg* is described in the time domain by the EQUATION block similar to (1d). Digital outputs are the number of all detections at the logical output port *counter* and an impulse at the port *impulse* if *lp_out* crosses the threshold. Logical input ports are *clk* and *reset*. The digital part

of the model is described between BEGIN PROCESS and END PROCESS with HDL-A similar to usual VHDL. This part corresponds to (1e).

```
ENTITY detector IS
  GENERIC(C, R:     REAL ;              -- input R, C
          fg:       REAL ;              -- cutoff frequency
          vs:       REAL);              -- threshold value
  PORT    (clk:     IN BIT;             -- clock
          reset:    IN BIT;             -- reset
          counter:  OUT REAL;           -- counter
          impulse:  OUT BIT);           -- threshold detection
  PIN     (inp:     ELECTRICAL);        -- analog input
END ENTITY detector;


ARCHITECTURE v1 OF detector IS
  STATE   lp_out:   ANALOG;             -- lowpass output
  SIGNAL  impulse_a,
          cross:    BIT;
  SIGNAL  counter_a:REAL;
BEGIN RELATION
  PROCEDURAL FOR INIT =>
    c := 1.0e-9;r := 1.0e4;
    fg := 1.0e3;vs := 1.0;              -- default values
    IF (c <= 0.0) OR (c <= 0.0) OR (fg <= 0.0) then
      REPORT "\n C,R, fg must be > 0.0 !"
      SEVERITY ERROR;
    END IF;
  PROCEDURAL FOR DC, TRANSIENT =>
    inp.i %= C*ddt(inp.v) + inp.v/R;    -- input current
  EQUATION (lp_out) FOR DC, TRANSIENT =>
    0.0== ddt(lp_out) + twopi*fg*lp_out - twopi*fg*inp.v;
  PROCEDURAL FOR AC =>
    REPORT "\n AC is not implemented !"
    SEVERITY ERROR;
END RELATION;
```

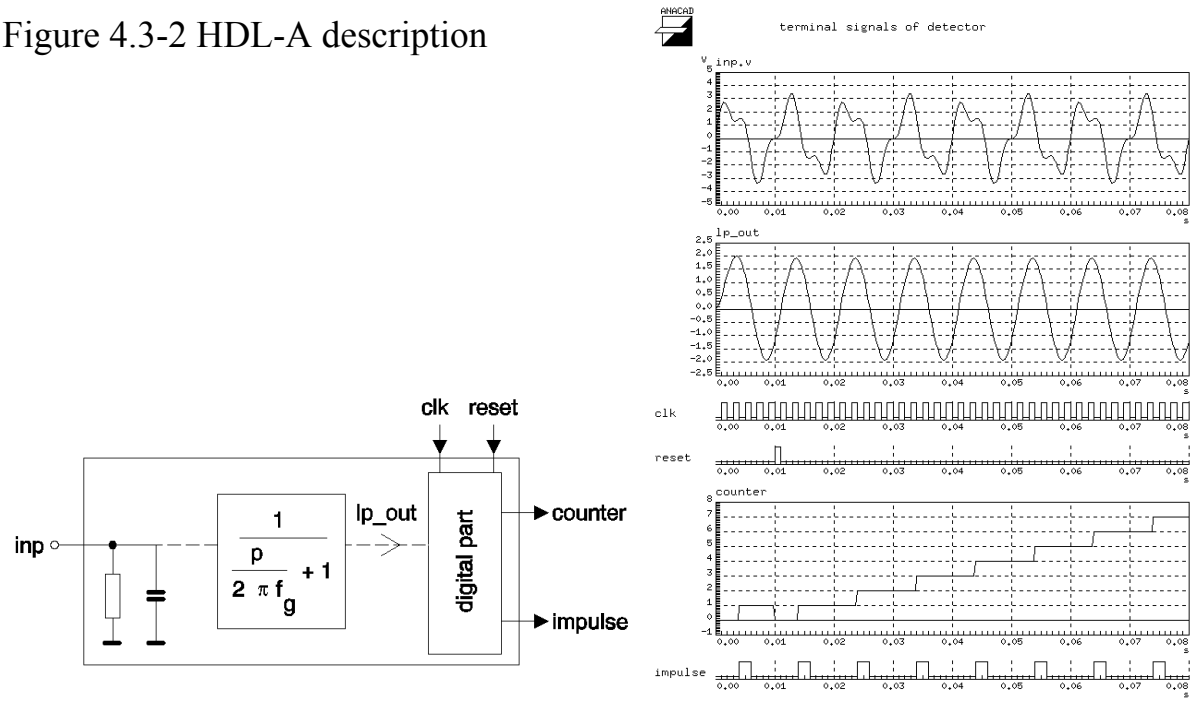
Figure 4.3-1 Circuit detector with an example for terminal signals

4.4 Current Possibilities of Analog Hardware Descriptions

A Language Reference Manual of VHDL-A is not yet available. So the question arises how to fill the gap between the different current behavioral modeling languages and the forthcoming VHDL-A capabilities in the next years.

```
PROCESS BEGIN
  counter <= 0.0;impulse <= '0';
  counter_a <= 0.0;impulse_a <= '0';
  cross <= '0';
  LOOP
    WAIT ON clk, reset;
    IF reset = '1' THEN
      counter <= 0.0;impulse <= '0';
      counter_a <= 0.0; impulse_a <= '0';
      cross <='0';                          -- rising clock edge
    ELSIF clk = '1' THEN                    -- lowpass output <
                                             threshold
      IF lp_out < vs THEN
        impulse_a <= '0'; cross <='0';
                                            -- lowpass output >=
                                             threshold
      ELSIF cross = '0' THEN                -- first crossing
        counter_a <= counter_a + 1.0; impulse_a <= '1';
        cross <= '1';
      ELSE
        impulse_a <= '0';
      END IF;
    END IF;
    -- assignment of auxiliary signals to digital outputs
    counter <= counter_a; impulse <= impulse_a;
  END LOOP;
END PROCESS;
END ARCHITECTURE v1;
```

Figure 4.3-2 HDL-A description



Today most macromodels (analog and some mixed analog-digital) in the libraries of IC vendors and users are constructed as SPICE circuits and/or are expressed in powerful well-established HDLs, e.g. MAST, FAS. Both behavioral languages have the widely used capability to include C programs. Many other simulators have C interfaces, too, but they all use different modeling strategies, interface descriptions, and function libraries (to access simulator-internal data such as *last_simulation_time* or *actual_time_step*). The inclusion of C texts will also be allowed in VHDL-A, but the details are not yet clear. Many designers are interested in the re-usability of their macromodels and C programs. For analog modeling a C-based behavioral modeling language as a SPICE extension is discussed [21, 22].

Therefore, an additional way of macromodeling should be investigated:

a) Use the group of equations (1) as a „standardized“ mathematical behavioral description, especially for analog submodels,

b) Give the model description a VHDL-like structure:
   - external view at the component (similar to the ENTITY part),
   - declaration of additional variables and description of the terminal behavior with equations (similar to the ARCHITECTURE part),

c) Express the behavioral part in „simulator-neutral“ C code,

d) Fit the programmed macromodel to the simulator and the structural description language in use.

For structural descriptions common netlist formats such as SPICE, EDIF, Verilog, or VHDL may be used. Simple mixed-signal macromodels may be developed as pure behavioral models. Other macromodels may be combined behavioral/ structural models, e.g. with an analog behavioral part and a very large gate level circuit or a register transfer model.

This method is not in contradiction to the VHDL-A standardization because these behavioral models may be easily transformed for being used in VHDL-A simulators as soon as they are on the market. Of course, this method has not the power of full VHDL-A, though it offers a simple way to achieve re-usability of a large amount of code implemented in today´s macromodel libraries and for increasing the efficiency of SPICE-related simulators with interfaces to C.

5. CONCLUSIONS

The construction of mixed-signal macromodels as a combination of analog and digital, behavioral and structural submodels has proved very useful.

Further impulses will arise from

- improvements in the design process (top-down design, stepwise refinements of system models)
- verification (model updating, back-annotation)
- fault simulation and fault modeling
- nonelectrical systems
- coupling of simulators.

The main drawback of the heuristic approach is the man-power needed for constructing the models and determining the model parameters: days, weeks, or months have to be spended. Therefore, one important task will be the increase of automation in the modeling strategy.

6. ACKNOWLEDGEMENT

The paper presents results of our group „Modeling and Simulation“ at FhG-IIS/EAS Dresden. In particular, the work of Dr. Christoph Clauss and Dr. Peter Trappe is included in this presentation. We gratefully acknowledge the cooperation with our partners at Siemens EZM Villach (Dr. F. Dielacher, R. Petschacher, H. Zojer), Siemens München (Dr. Müller-Liebler, Dr. R. Koch), TEMIC Heilbronn (M. Hecker, Dr. R. Vahrmann), TEMIC Ulm (H. Golberg), and other designers from these companies. We also thank D. Backhaus (ANACAD) for the opportunity to work with an preliminary HDL-A version of the simulator ELDO. We would like to thank A. Schneider and S. Wünsche for their support in preparing this paper.

The work was partially supported by the German Ministry for Research and Technology (BMFT) within the JESSI AC 12 Project and the Joint Research Project „Modellbibliothek für komplexe analoge Bauelemente (BIBLIO)“.

REFERENCES

[1] Brown, R.W.: Device models for circuit analysis programs. Computer Aided Design (1969)1, 33-42

[2] Greenbaum, J.R.: Make linear models of op amps. Electronic Design 17(1969)19, 92-96

[3] Saleh, R. A.; Newton, A. R.: Mixed-Mode Simulation. Boston - Dordrecht - London: Kluwer Academic Publisher, 1990.

[4] Vlach, M.: Modeling and Simulation with Saber. 3rd Annual ASIC Seminar and Exhibition, Rochester 1990, T-11.1 - T-11.9

[5] El Tahawy, H.; Rodriquez, D.; Garcia-Sabiro, S.: Mayol, J.-J.: VHD$_e$LDO: A new Mixed Mode Simulation. Proc. EURO-DAC 1993, Hamburg, 546 - 551.

[6] Desclives, C.: SMASH: a Mixed-signal Multi-level Simulator. EDAC-EUROASIC, Paris, February 1993

[7] Clauß, C.; Haase, J.; Schwarz, P.; Trappe, P.: Makromodellierung gemischt analog-

digitaler Schaltungen. 2. GME/ITG-Diskussionssitzung "Entwicklung von Analogschaltungen mit CAE-Methoden", Ilmenau, März 1993, 155 - 160.

[8] Goedecke, M.; Hamad, H.; Huss, S.: A Methodology for the Development of System-Level Simulation Models for Analog Functional Blocks. Archiv Elektronik u. Übertragungstechnik 49(1995)2, 72-80.

[9] Gerbershagen, M.: CLANG - Ein Programm zur prozeduralen Simulation und Charakterisierung analoger Zellen. Techn. Report, TEMIC / AEG Ulm, 1992

[10] Schwarz, P.: KOSIM - A multi-level simulator for the design of integrated circuits. In: Reichl, H. (Ed.): Proc. MICRO SYSTEM Technologies 90, Berlin: Springer-Verlag, 1990, 85-90

[11] Tiefenbacher, M.; Caldera, P.; Dielacher, F. et al.: A Four Channel CMOS Codec Filter Circuit SICOFI-4. IEEE J. SC-29(1994)8

[12] Zojer, B.; Koban, R.; Petschacher, R.; Sereinig, W.: Integration of a Subscriber Line Interface Circuit (SLIC) in a New 170V Smart Power Technology, Proc. ISPSD '95

[13] Straube, B.: Computation of the Terminal Behavior of Switch-level MOS Circuits. Techn. Report FhG-EAS Dresden, January 1995

[14] Vachoux, A.; Nolan, K.: Analog and Mixed-Level Simulation with Implications to VHDL. In: Mermet, J. (Ed.): Fundamentals and Standards in Hardware Description Languages. Dordrecht - Boston - London: Kluwer Academic Publisher, 1993, 281-330

[15] Bergé, J.-M.; Damon, D. G.; Vachoux, A.: VHDL Analog Extensions. Tutorial at EURO-DAC, Grenoble, September 1994.

[16] Saleh, R. A.; Rhodes, D. L.; Christen, E.; Antao, B. A.: Analog Hardware Description Languages. Proc. Custom Integrated Circuits Conf. CICC´94, San Diego, May 1994, 15.1.1 - 15.1.8.

[17] Visweswariah, C.; Chadha, R.; Chen, C.-P.: Model development and Verification for High Level Analog Blocks. Proc. 25th DAC 1988, 376 - 382.

[18] Casinovi, G.; Yang, J.-M.: Multi-Level Simulation of Large Analog Systems Containing Behavioral Modells. IEEE Trans. CAD 13 (1994)11, 1391 - 1399.

[19] Clauß, C.; Haase, J.; Kurth, G.; Schwarz, P.: Extended Amittance Description of Nonlinear n-Poles. Archiv Elektronik u. Übertragungstechnik 49(1995)2, 91-97.

[20] Haase, J.: Computation of Transfer Characteristics of Multivalued Resistive Nonlinear Networks. Summer Symp. Circuit Theory, Prag 1982, 268 - 272.

[21] Desclives, C.: Analog behavioral modeling: natural or not? Dolphin Res. Note, 1994

[22] Gross, C.: La modelisation C vient aux simulateurs SPICE. Electronique No. 45, Fevrier 1995