Mixed-Signal Event-Driven Simulation of a Phase-Locked Loop

Martin Schubert[1]

Abstract

The mixed-signal event-driven (MixED) simulation algorithm using standard VHDL is capable of modeling a number of analog and mixed-signal problems in digital circuits, e.g. RCL networks representing pads or wires, charge pumps, dynamic logic, voltage-controlled oscillators, phase-locked loops, etc. Important features are single-kernel simulation as well as rapid A/D and D/A interfacing. This paper demonstrates the simulation of a phase-locked loop (PLL), which is one of the most interesting applications of the MixED method. Many digital designs contain a PLL as only mixed-signal building block. The MixED method allows to simulate such designs with standard VHDL.

Index terms -- *mixed-signal event-driven simulation, standard VHDL, analog models, PLL*

1 Introduction

A method for the simulation of analog and mixed-signal circuitry using standard VHDL was recently presented [1,2,3]. Due to convergence problems, this method is applied to the simulation of logic devices and some analog building blocks as charge pumps of phase locked loops.

There are two major advantages of the mixed-signal event-driven (MixED) method compared to most other mixed signal simulators:

1. Due to single kernel simulation the MixED method allows for fast D/A and A/D interfacing.
2. Event-driven modeling allows the computer power to be focused on specific locations also in the analog part of the circuit, because adaptive time stepping is performed individually for every node.

Any mixed analog-digital simulator such as Spice [4], Verilog-AMS or VHDL-AMS [5,6] must introduce specific models to interface the analog and digital simulation kernel. To meet the challenges of mixed-signal design most software vendors offer a co-simulation method using a backplane to interface analog and digital data and events. The analog and digital kernels use the CPU alternately, where one has to precede the other. This requires the saving of a lot of information or the implementation of techniques such as "backtracking" at the cost of computational efficiency.

The MixED module is coded in standard VHDL [1,2,3]. The user has to instantiate one initialization component to compute some parameters (e.g. the simulator's actual resolution) that are global to the module. MixED devices are VHDL models that communicate via nodes with conservative behavior to simulate non-digital devices. The basic driver model is the controlled source element with resistive output impedance illustrated in Fig. 1.1(c). All other MixED devices are based on this controlled resistive source model.

Fig. 1.1 illustrates the translation of a CMOS inverter into a resistive source. MixED models available so far are controlled sources with resistive, inductive, capacitive or mixed resistive-capacitive output impedance, as well as Spice level-1 MOSFET models assuming given bulk and source voltages, furthermore inverters and a voltage controlled oscillator (VCO). As MOSFET models are non-linear, they require some iterations. For realistic cases with resistive or capacitive loads only a few iterations are necessary to obtain convergence with reasonable accuracy.

Digital networks are assumed to be chains of unidirectional controlled sources, whereas several sources may drive a node.

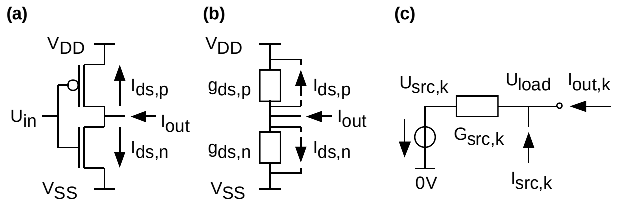

Figure 1.1: (a) CMOS inverter, **(b)** equivalent circuit representation, **(c)** controlled source representation.

[1] The author is adjunct to Fachhochschule Regensburg, Fachbereich Elektrotechnik, Seybothstr. 2, D-93053 Regensburg, Germany. Email: martin.schubert@e-technik.fh-regensburg.de.

2 Timing Considerations

Proper time axis treatment is a basic skill of any transient simulator. As the MixED module is based on standard VHDL it has to employ the standard VHDL time axis. However, practical simulators impose additional constraints.

VHDL time points are by definition integral multiples of the minimum time step Δt_{min}, which is defined to be 1 fs in the VHDL language reference manual (LRM). Practically, this rule would constraint the usable time axis to a maximum of

$$t_{max} = \text{TIME'HIGH} \cdot \Delta t_{min} = 2^{31}\, \Delta t_{min} = 2.147\ \mu s$$

for simulators with so-called '32-bit timing' or

$$t_{max} = \text{TIME'HIGH} \cdot \Delta t_{min} = 2^{63} \cdot \Delta t_{min} = 2{:}33{:}43\ h$$

for simulators with so-called '64-bit timing'. The exponent in the formulae above is lowered by one because `TIME'HIGH=-TIME'LOW` by definition.

Most VHDL simulators allow the user to select larger values of Δt_{min} to obtain correspondingly larger values of t_{max}. Therefore, the user cannot rely on a resolution of 1 fs in practical applications. Taking this practical aspect into account, an initialization procedure of the MixED module checks for the simulator's actual resolution during the first delta at time=0 sec. This information is then assigned to one of the signals which are global to the MixED module.

A circuit modeled with the MixED module is a set of devices realized by component instantiations. There is no central control, that could compute a time step for all MixED devices. This devices communicate with their neighbors via so-called "mixed nodes", having the module specific data type `t_node`. As the event-driven nature of VHDL is preserved, mixed nodes perform time stepping independently from each other. Any device attached to a node can trigger a time point for this node. The intelligence of time step computation is coded individually into the different MixED devices, allowing for a very flexible behavior optimized to the particular situation. This is necessary, because a module based on standard VHDL cannot step back in time.

This simple fact implicates a trade-off between simulation speed using large time steps and accuracy considerations recommending small time steps. This issue will be explained more detailed for the A/D and D/A conversion models in the next chapters.

3 Digital-to-Analog Conversion

D/A conversion seems to be a simple task on the first glance. Assuming V_{DD} and V_{SS} to be the power supply voltages, D/A conversion could be done using the following peace of VHDL code:

```
SIGNAL logic_value:BIT;
SIGNAL real_value: REAL;
...
real_value <= Vss WHEN logic_value='0'
          ELSE Vdd;
```

Listing 3.1: Piece of VHDL code performing very simple D/A conversion.

The key problem, however, is a correct interpretation of events on the time axis rather than number conversion.

Fig. 3.1(a) illustrates the problem: VHDL assumes abrupt signal changes (events) in exactly defined time points and constant signal values between them. The MixED method takes the last time point into consideration when computing the slope of an edge. Due to the implicit backward Euler integration method employed, the slope computed between t_1 and t_2 in Fig. 3.1(a) depends on the time step $\Delta t = t_2 - t_1$. In other words: The simulator makes assumptions about the past between t_1 and t_2 when arriving at t_2. Signal value and slope are unknown for $t > t_2$ before the solution at t_3 is computed. Therefore, the simple A/D conversion method shown in listing 3.1 and Fig. 3.1(a) is unsuitable for well controlled digital-to-analog edge translation.

Fig. 3.1(b) illustrates an improved A/D edge conversion scheme. In response to a digital event the respective analog signal sets a time point with the old signal value, then proceeds for a minimum time step (Δt_{min}) and sets a new time point with the new signal value. Fig. 3.2(a) illustrates the required input and output data of such an A/D converter.

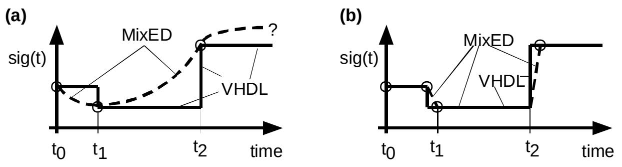

Figure 3.1: Generation of fast analog edges:
(a) The MixED method refers to the last time point. **(b)** Introduction of time points to obtain fast edges.

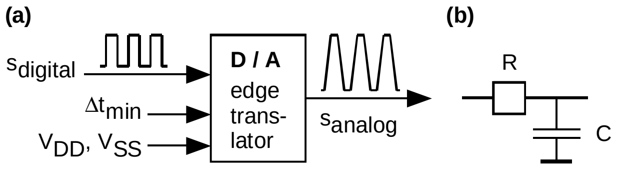

Figure 3.2: A/D conversion of digital edges
(a) using minimum possible rise and fall times
(b) appending an RC element if necessary.

This method produces unrealistic fast edges depending on the setting for Δt_{min}. If this is undesirable, e.g. an appended RC low pass can solve the problem. R may be time dependent. Vogelsong [7] proposed a "do-it yourself approach to propagation waveshaping" by explicitly defining the output waveshape in response to the input event using predefined output wave functions. Several function proposals can be found in [8]. However, the advantage of saving one or some analog nodes is paid for with (1) a lot of error prone coding, (2) numerous analytic problems when considering a series of short random pulses and (3) it forces the user to adjust additional - often very abstract - data.

4 Analog-to-Digital Conversion

4.1 General Considerations

A/D conversion is particularly critical with respect to numerical noise. A very simple but noise generating method of A/D conversion is shown in listing 4.1.

```
SIGNAL logic_value:BIT;
SIGNAL real_value, Uref: REAL;
...
logic_value <=
  ’0’ WHEN real_value < Uref ELSE
  ’1’;
```

Listing 4.1: Piece of VHDL code performing very simple A/D conversion introducing numerical noise.

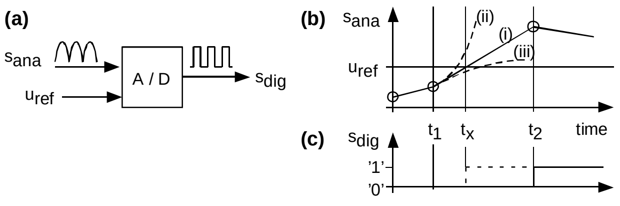

Figure 4.1: **(a)** A/D converter,
(b) analog input signal s_{ana} and speculated progress,
(c) digital output signal s_{dig}.

The accuracy of this method is limited by the time step width as can be seen from Fig. 4.1: While the solid line (i) in Fig. 4.1(b) crosses u_{ref} at $t_x<t_2$, the above A/D conversion statement reacts at t_2, as indicated by the solid line in Fig. 4.1(c). Remembering that time the points of a mixed node are triggered by the devices connected to the node, it is the task of the A/D converter to estimate the cross-over time point t_x when being at t_1. This estimation is not simple, because the curve may be nonlinear as indicated by the dashed lines (ii) and (iii) in Fig. 4.1(b). To minimize the risk of oversized time steps, the A/D converter may compute an approximated t_x and then schedule a time point at $t_1+\alpha\cdot(t_x-t_1)$, with the fitting parameter α having a typical value of $0<\alpha\le 1$.

4.2 VCO with digital output signal

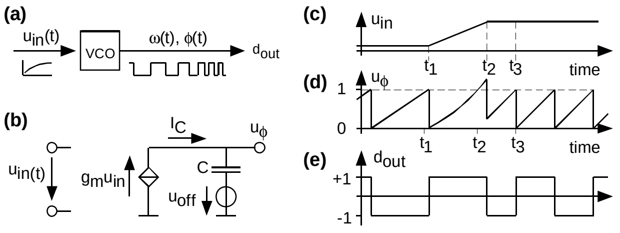

Figure 4.2: **(a)** Digital VCO operating as A/D converter, **(b)** realization of the VCO: capacitor C as integrator, **(c)** analog input signal u_{in}, **(d)** voltage u_ϕ modeling the phase **(e)** digital output signal d_{out}.

A specific kind of A/D converter is the voltage controlled oscillator (VCO) shown in Fig. 4.2(a): It receives an analog input voltage $u_{in}(t)$ as sketched in Fig. 4.2(c) and delivers a digital output signal d_{out} as illustrated in Fig. 4.2(e). Fig. 4.2(b) shows a behavioral model of the VCO using a voltage controlled current source charging the capacitor C. The voltage u_ϕ models the phase, which may be decremented as shown in Fig. 4.2(d) for periodic functions. In Fig. 4.2 $\Delta u_\phi=1$ corresponds to a half wavelength of d_{out}.

The VCO was modeled to deliver the output frequency

$$f_{VCO}(t) = K_{VCO} \cdot u_{in}(t) + f_{FR}$$

with the differential input voltage $u_{in} = u_{inp} - u_{inn}$, the VCO amplification K_{VCO} and the free-running frequency f_{FR}. A PLL requires phase information, which is obtained from integration over frequency:

$$\phi_{VCO} = \int \omega_{VCO} dt = 2\pi K_{VCO} \int \left(u_{in}(t) + f_{FR}\right) dt$$

This integration is realized with the capacitor C in Fig. 4.2(b).

The characteristic feature of periodic functions is repetition, e.g. sin(2πx) = sin(2π(x-n)) with n being any integral number. In the circuit of Fig. 4.2(b) the phase is modeled by the voltage $u_\phi = \pi \cdot \phi_{VCO}$, where u_ϕ is obtained by integration of current in the capacitor C. u_ϕ is decremented by 1 when $u_\phi \geq 1$. The output signal d_{out} of data type `BIT` is inverted if a decrementation of u_ϕ occurs. To avoid numerically generated phase noise, the situation $u_\phi > 1$, as sketched near t_2 in Fig. 4.2(d), should be avoided. This requires accurate prediction of the time point when point $u_\phi = 1$.

5 Application: Phase-Locked Loop Model

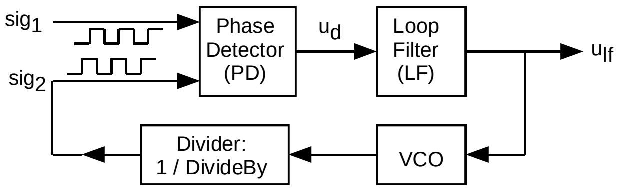

Figure 5.1: Digital phase-locked loop (DPLL)

The digital PLL (DPLL) is a frequently required building block for otherwise digital designs. If the analog and mixed-signal modeling requirements of a DPLL can be met with standard VHDL, a large number of designs can be done without VHDL-AMS. Even a non accurate PLL model running on a standard VHDL simulator can be valuable for the verification of the digital part of the circuit, e.g. to check wiring and phase noise tolerance.

Fig. 5.1 shows the block diagram of a DPLL. Fig. 5.2 illustrates a realization using a phase-frequency detector (PFD). Its advantage compared to a "normal" phase detector (PD) is its capability of frequency detection if the loop is unlocked. With a PFD the VCO can be controlled without other means of locking.

D/A conversion is done with minimum width edges in Fig. 5.2, assuming that the width of the digital edge is insignificant. For the simulations shown in Figs. 5.3 and 5.4 Spice level-1 MOSFET models were employed:

M_N: V_{Tn} = 0.62 V, $ß_n = 781\ \mu A/V^2$, $\lambda_n = 0.2$
M_P: V_{Tp} = -0.61 V, $ß_p = 779\ \mu A/V^2$, $\lambda_p = 0.2$

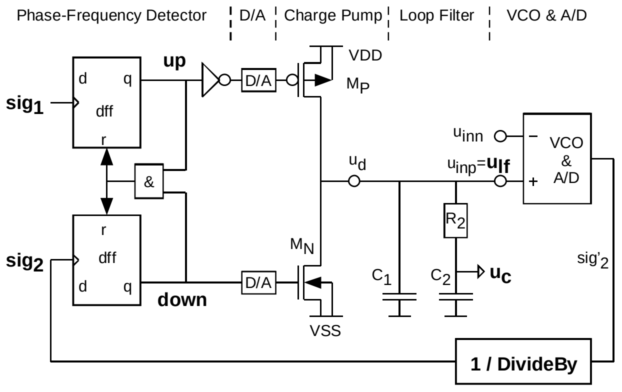

Figure 5.2: Realization of the DPLL sown in Fig. 5.1.

The data for the loop filter was taken from data sheets for mobile phones:

C_1 = 336 pF, R_2 = 6.5 KΩ, C_2 = 6.72 nF.

Typically, u_C and not u_{lf} is used to control the VCO in Fig. 5.2. This requires some minor extensions to the model but is no fundamental problem of the MixED method. V_{DD}=5V and V_{SS} = 0V were chosen too high to cover the frequency range with the given K_{VCO}.

All simulations were performed on a desktop computer with a 200 MHz Pentium II compatible AMD CPU, operated under Windows NT 4.0 with 96 MB RAM. Simulation and graphics were done using the "ModelSim PE/Plus Version 4.7h" simulator of MTI [9]. The simulator's resolution was set to 1 fs.

Fig. 5.3 shows a simulation of the first 150 us of the PLL with an initial filter voltage of $u_{lf}=u_C$=0V. Externally given is the frequency f_1 = 200 KHz of signal sig_1 with a transparent divider using `DevideBy`=1. VCO parameters: f_{FR} = 200 KHz, K_{VCO} = 4.5 KHz/V, U_{inn} = 3V. This simulation consumed about 10 seconds of CPU time on the computer described above.

Fig. 5.4 shows a simulation from 0-1.2ms, which is close to the requirements of mobile phones. Externally given is the frequency f_1 = 200 KHz. The divider was set using the following VHDL statement:

```
DevideBy <= 4450, 4448 AFTER 200 us,
4300 AFTER 500 us, 4600 AFTER 800 us;
```

The VCO was set to the following parameters: f_{FR} = 890 MHz, K_{VCO} = 20 MHz/V, U_{inn} = 3V. The simulation of the 1.2 ms consumed nearly 3 hours (exactly 2:53:55) of CPU time on the computer described above.

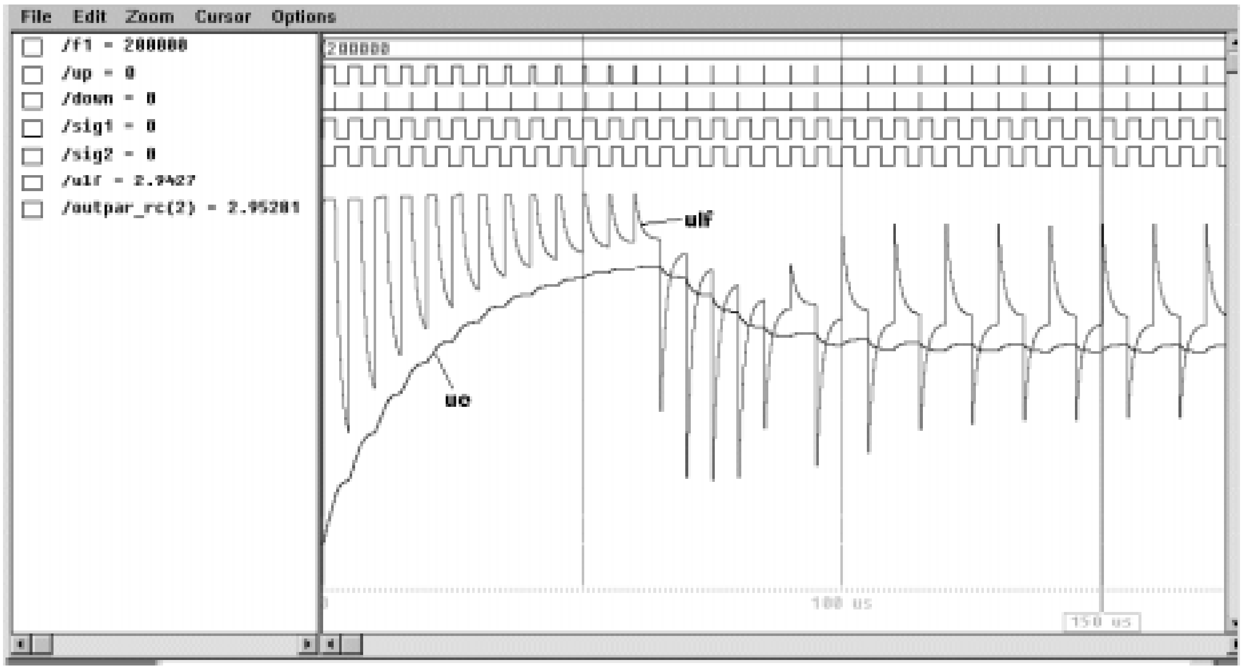

Figure 5.3: Screen shot of a DPLL simulation according to Figs. 5.1 and 5.2. The graphic contains a shift between u_{lf} and u_C for better visibility: The voltage u_{lf} strives exponentially versus u_C if both MOSFETs are high impedant and should therefore be centered around u_C.

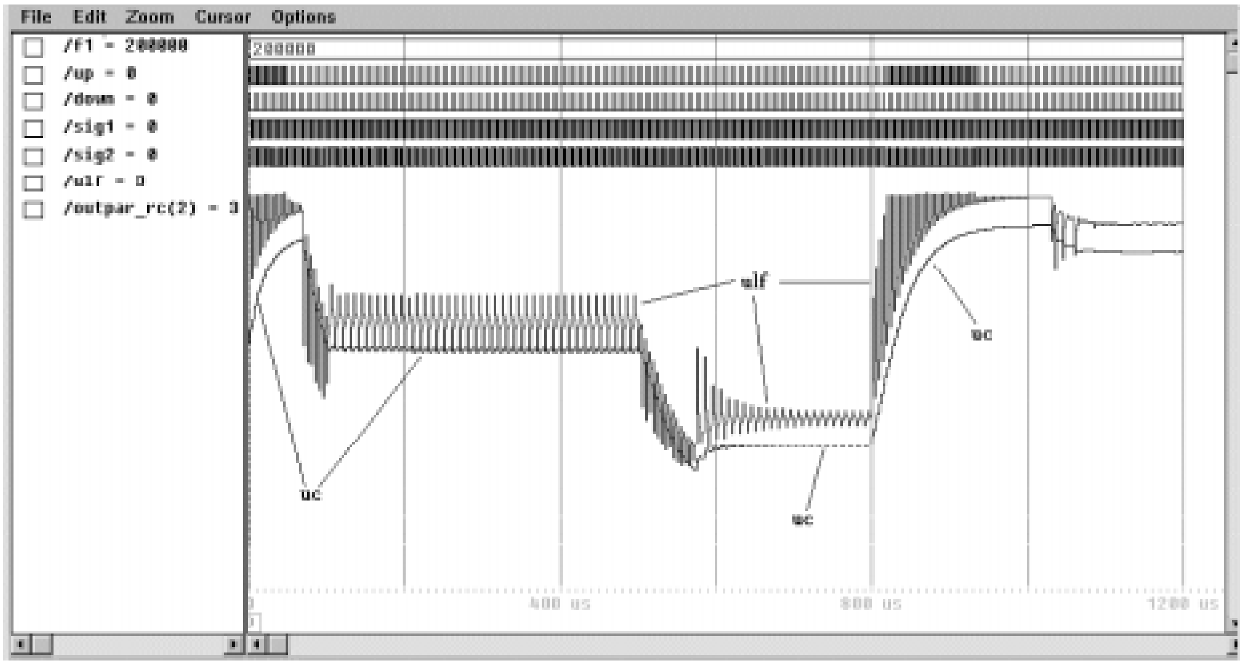

Figure 5.4: Screen shot of a DPLL simulation according to Figs. 5.1 and 5.2. The graphic contains a shift between u_{lf} and u_C for better visibility: The voltage u_{lf} strives exponentially versus u_C if both MOSFETs are high impedant and should therefore be centered around u_C.

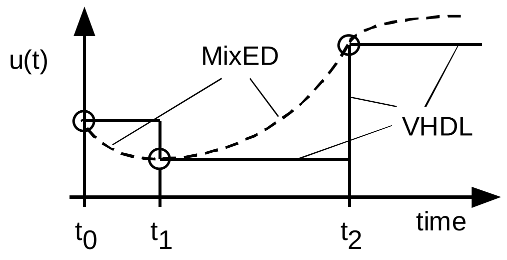

Figure 5.5: Differences between digital an analog interpretation of graphical information.

The interpretation of the screen shots shown in Figs. 5.3 and 5.4 has to take into account the limitations of a digital simulator used for analog simulations. Fig. 5.5 illustrates the assumptions of VHDL (solid line) with abrupt changes in given time points on the one hand and an analog curve (dashed line) interpolated between the data given at the time points t_i.

An interpolation algorithm should take into account the charge conserving model assumptions for capacitors proposed by Yang, Epler and Chatterjee [10] which deliver $u(t_i)$, $i_C(t_i)$ and $du(t_i)/dt$.

Due to a lack of behavioral modeling capabilities it is difficult to create a Spice model that allows for a fair comparison between the MixED method and Spice. Respective Spice models tried out by the author are most probably more accurate but seem to be much slower. VHDL-AMS is currently not available to the author.

6 Summary

A method allowing for a limited amount of analog simulation using standard VHDL was applied to digital and mixed-Signal circuits. Single kernel simulation and fast A/D and D/A interfacing makes it suitable for simple analog problems with massive interaction to surrounding digital circuitry.

A model for a digital phase-locked loop was presented using standard VHDL. Timing constraints and issues of A/D and D/A conversion were discussed. Though the model is not yet tested versus experimental data, it was proved that a behavioral DPLL model using standard VHDL is possible.

The model works with reasonable CPU power consumption, so that it can be simulated together with other digital circuitry on a standard VHDL simulator. Even if the model is not yet sufficiently accurate, it significantly extends the design verification possibilities for standard VHDL tools.

7 Acknowledgments

The author would like to thank Prof. Dr. Gareth Monkman of the Fachhochschule Regensburg for reading the manuscript and helpful advice.

8 References

[1] Schubert, M.; “Mixed Analog-Digital Signal Modeling Using Event-Driven VHDL“, X Brazilian Symp. on Integrated Circuit Design - SBCCI’97, Porto Alegre, Brazil, Aug. 25-27, 1997.

[2] Schubert, M.; “Operational Amplifier Modeling Using Event-Driven VHDL“, IEEE/VIUF International Workshops on Behavioral Modeling and Simulation - BMAS’97, Washington D.C., USA, October 20-21, 1997.

[3] Bartsch, E.; Schubert, M.; “Mixed Analog-Digital Circuit Modeling Using Event-Driven VHDL“, IEEE/VIUF International Workshop on Behavioral Modeling and Simulation - BMAS’97, Washington D.C., USA, October 20-21, 1997.

[4] OrCAD: Pspice, Internet Address: http://www.orcad.com/products/pspice/.

[5] Christen, E.; Bakalar, K.; “VHDL 1076.1 - Analog and Mixed-Signal Extension to VHDL“, Europ. Design Autom. Conf. (EuroDAC’96) with EURO-VHDL’96, Geneva, Switzerland, Sept. 16-20, 1996, pp. 556-561.

[6] Vachoux, Alain; "Analog and Mixed-Signal Extensions to VHDL", Analog Integrated Circuits and Signal Processing Journal, Vol.16, pp. 185-200, Kluwer Academic Publishers, June 1998.

[7] Vogelsong, R.; "Analog Macro/Behavioral Modeling Techniques", IEEE/VIUF International Workshops on Behavioral Modeling and Simulation - BMAS'98, Orlando, Florida, USA, October 27-28, 1998

[8] M. Schubert, B. González, "Feature Oriented Analog Waveform Behavioral Modeling", IEEE/VIUF International Workshops on Behavioral Modeling and Simulation - BMAS'98, Orlando, Florida, USA, October 27-28, 1998.

[9] Model Technology Incorporation, Internet address: http://www.model.com, Email address: support@model.com.

[10] P. Yang, B. D. Epler, P. K. Chatterjee, "An Investigation of the Charge Conservation Problem for MOSFET Circuit Simulation", IEEE Journal of Solid-State Circuits, Vol. SC-18, No. 1, Feb. 1983.