SEVENTH FRAMEWORK PROGRAMME
THEME [ICT-1-3.1]
Next-Generation Nanoelectronics Components and Electronics Integration

Deliverable Report

Work Package 2 – IC buffers' innovative modelling approach

Deliverable D2.1- Report describing the structure of the new parametric model for digital ICs and detailed procedure for model validation.

Contract no.:	216732
Project acronym:	MOCHA
Project full title:	MOdelling and CHAracterization for SiP Signal and Power Integrity Analysis
Start date of project:	1st January 2008
Duration:	2 years
Instrument:	Small or medium-scale focused research project - STREP
Report responsible partner:	Politecnico di Torino; Instituto de Telecomunicações
Report preparation date:	22/10/2008
Dissemination level:	PU
WP2 leader:	Igor Stievano
WP2 leader organization:	Politecnico di Torino
Project coordinator:	Antonio Girardi
Project coordinator organization:	Numonyx Italy Srl

Revision: 1.0

Content

1. Introduction

Nowadays, the models of I/O buffers of digital Integrated Circuits (ICs) are a key resource for the simulation of the signal integrity and electromagnetic compatibility effects of digital circuits. Buffer macromodels are usually based on simplified equivalent circuits and the Input Output Buffer Information Specification (IBIS) [1]. Models based on IBIS data established as a standard widely used, are supported by manufacturers and accepted by most Electronic Design Automation (EDA) tools. Recently, other approaches to IC buffer macromodels, that supplement the IBIS resource and provide improved accuracy, have been proposed [2, 3, 6]. These approaches are based on system identification methods and parametric relations. They exploit a mathematical description of current and voltage evolution at the buffer port, possibly reproducing complicated dynamic nonlinear behaviors of the modeled devices. Of course, macromodels obtained by these parametric approaches remain almost as efficient as macromodels based on IBIS data and can be easily included in EDA tools as SPICE subcircuits or VHDL-AMS/Verilog-A descriptions.

The aim of Task 2.1 of WP2 is to extend the modeling domain of buffer model. Present macromodels, can allow only for limited variations of the power supply voltage, on the order of 10% of the nominal power supply voltage. For usual simulation problems, this limitation does not affect the accuracy of predictions. In fact, most application exploit single dye ICs and Printed Circuit Boards (PCBs) with decoupling caps and power planes that ensure small variations of the supply voltage. On the other hand, recent applications, like the stacked System in Package (SiP) devices, suffer from larger variation of the supply voltage and demand for buffer macromodels allowing for such large variations.

An example of the effects of supply voltage variations on macromodel accuracy is given in Fig. 1. The waveforms of this Figure are the responses of the buffer output and supply pin voltages for a device sending a pseudo-random bit sequence on a PCB interconnect. The solid line curves are the reference responses obtained by using a detailed device-level model of the buffer, whereas the dashed ones are the responses predicted by a state-of-the-art model of the buffer [2]. The current absorption and the impedance of the power supply network of this example are such that the supply voltage fluctuations are on the order of 40% of nominal power supply voltage. The response predicted by the model is affected by significant distortion and a timing error as large as the 10% of the bit time, thereby proving that so large variations of the supply voltage are outside the validity domain of the model. Recently, preliminary studies on the extension of the current models based on IBIS to account for the large fluctuation of the supply voltage have been reported by ST microelectronics and Intel. A proposal based on these studies, that is named “Gate Modulation Effect”, has been accepted and included in the current ver. 5.0 of the IBIS specification [1]. More details can be found in the Buffer Issue Resolution Documents (BIRDs) [4, 5] and in the paper [7]. Here we elaborate on these results to achieve the required modeling domain extension.

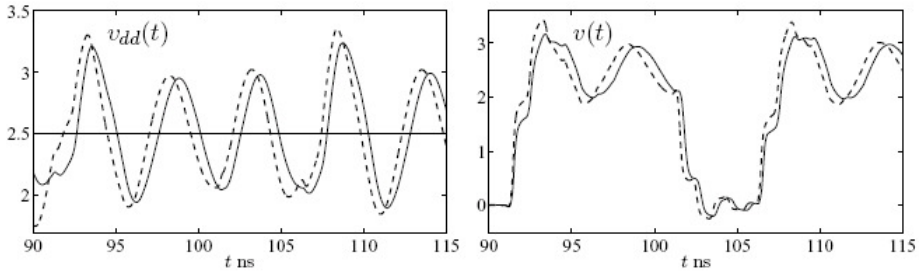

Figure 1: Output and supply port voltage waveforms (v(t) and vdd(t), respectively) for the port of a digital IC energized by a real power supply network and driving a PCB interconnect (see text). Solid lines: reference; dashed lines: macromodel [2].

2. Buffer models

This Section briefly reviews the structure of macromodels for IC output buffers. For the sake of simplicity, the discussion is based on the output buffer of a single-ended device, whose structure is shown in Fig. 2. State-of-the-art macromodels for this structure are based on the following two-piece relation [1, 2].

$$i(t) = w_H(t)\, i_H(v(t),v_{dd}(t),d/dt) + w_L(t)\, i_L(v(t),v_{dd}(t),d/dt) \quad (1)$$

where i_H and i_L are submodels accounting for the device behavior in the logic high and low state, respectively, and the time-varying functions $w_H(t)$ and $w_L(t)$ provide the transition between the two submodels, i.e., the switching between the two logic states. A similar equation holds for the power supply current i_{dd}. Submodels i_H and i_L can be obtained from either simplified equivalent circuit representations (e.g., see IBIS [1]) or identification methods and parametric relations [2]. Present models (like the one used in the example of Fig. 1) are based on the simplifying assumption that the variations of the supply voltage are small. With this assumption, the weighting functions $w_H(t)$ and $w_L(t)$ are independent of the v_{dd} variable and the static characteristics of submodels i_H and i_L are those holding for the nominal value of v_{dd}.

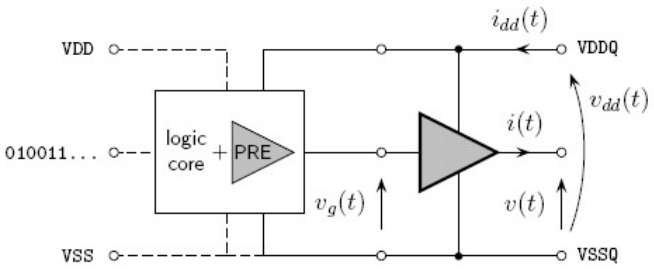

Figure 2: Structure of the output buffer of a digital integrated circuit with its relevant electrical variables. The bold triangle symbol indicates the last inverter stage and vg its input voltage.

3. Extended buffer models

This Section extends the basic model structure (1) to account for the large variations of the power supply voltage v_{dd}. For conciseness, the analysis focuses on models based on parametric relations only, as discussed in [2], where the submodels i_H and i_L are splitted into the sum of a static and a dynamic contribution (see the Appendix of [3]). The argument developed here, however, can be applied to any macromodel based on equation (1). The splitted structure for the submodel i_H writes

$$i_H(t) = i_{sH}(v_{dd}(t),v(t)) + i_{dH}(v(t),v_{dd}(t),d/dt) \quad (2)$$

where i_{sH} is the static surface of the output current of the buffer at high output state and i_{dH} is a parametric model accounting for the nonlinear dynamic behavior of the output current. A similar equation holds for $i_L(t)$. The extension of macromodels is illustrated for an example device that is a 8-bit bus transceiver with four independent buffers (model name SN74ALVCH16973, power supply voltage VDDQ = 2.5 V). The reference responses of this device are obtained by its HSPICE device-level model and are used for both the estimation of the parameters of our macromodel and for the assessment of its performance. The static output characteristics of the example buffer are plotted in Fig. 3 for both logic states and for different v_{dd} values. In this Figure, v_{dd} is varied within the range [-30%,+30%] of VDDQ. Present models use $i_{sH} = \underline{f}_{sH}(v_{dd} - v)$ and $i_{sL} = f_{sH}(v)$, where f_{sH} and f_{sL} are the

static characteristics of the output port current at high and low logic state for the nominal power supply voltage VDDQ. In other words, present models identify i_{sH} and i_{sL} with the solid thick curves of Fig. 3, that is a rough approximation when the supply voltage variation is large. The proposal of [5] suggests to approximate the complete output port current surface by means of the following formula

$$i_{sH}(v,v_{dd}) = k_H(v_{dd})\, f_{sH}(v_{dd}-v);\; i_{sL}(v,v_{dd}) = k_L(v_{dd})\, f_{sL}(v_{dd}-v) \qquad (3)$$

where coefficients k_H and k_L account for the variation of the supply voltage v_{dd} and are computed by matching the approximation to the actual surface for v = 0 and v = VDDQ (see the vertical gray lines of Fig. 3). It is worth noting that, as any approximation defined along a line, this approximation is exact at v = 0 and v =VDDQ only. In actual operation, however, the domain of the output characteristic explored by i and v variables does not concentrate on v = 0 and v = VDDQ, because i and v keep close to the buffer load lines. This property is demonstrated in Fig. 4, where some typical i(t) and v(t) trajectories are drawn on the static curves of Fig. 3. The plotted trajectories are obtained from the responses of the example device driving a transmission line with characteristic impedance $Z_0 = 50\Omega$. The trajectories develop along the load lines of the transmission line (gray straight lines of Fig. 4) and v(t) is hardly close to the supply rails shown in Fig. 3. In order to account for this effect, we use (3) with k_H and k_L coefficients computed by matching the approximation and the real surface on the load lines of Fig. 4.

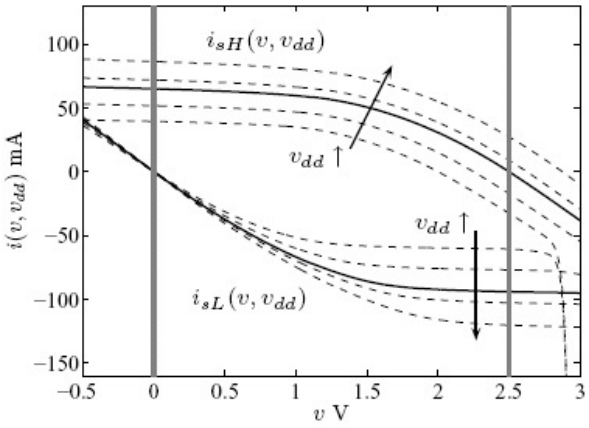

Figure 3: Solid-thick lines: DC output current of the example driver at fixed high and low output state and v_{dd} = VDDQ; dashed-thin lines: DC output current for different values of the power supply voltage vdd; vertical gray lines: power supply rails.

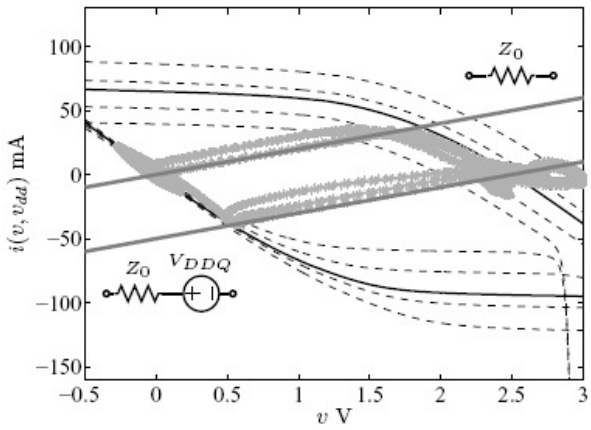

Figure 4: Solid-thick and dashed-thin lines: DC curves of Fig. 3; gray-line curves: samples of i(t) and v(t) waveforms for a 50Ω transmission line load; straight gray lines: load lines of the transmission line.

The inclusion of supply voltage effects in the static characteristics of (1), however, is not sufficient to obtain a model for large supply voltage variations. Most of the error visible in Fig. 1, in fact, comes from timing effects due to the supply voltage variations. In order to allow for timing variations, the dependence of the weighting functions $w_H(t)$ and $w_L(t)$ on vdd must be taken into account. In our model we identify the weighting function by combining functions obtained for different values of vdd (*e.g.*, 70%, 100% and 130% of VDDQ). More details on the computation of the weighting functions for constant supply voltages can be found in [2] and in the internal Milestones M2.1 and M2.2.

It is worth noting that the coefficients $k_{H,L}$ and the combination of the weighting function computed for different values of the power supply voltage can be obtained either the numerical simulation and real measurements.

4. Numerical results

In this Section, the accuracy of different models based on equation (1) is assessed by comparing their responses to the reference response obtained by the device-level model of the example buffer. In particular, the following models are considered: the state-of-the art model of [2] used for the example of Fig. 1 (model #1 in the following), and the models defined by (3) with k_H and k_L coefficients computed either by matching on power supply rails (model #2) or by matching on the load lines of Fig. 4 (model #3). For both model #2 and model #3, the weighting functions w_H and w_L of (1) are parameterized on vdd as outlined in the previous Section. All the models have been implemented in SPICE. The transient responses predicted by the models for a circuit composed of the example buffer sending the '010' sequence on an open-ended ideal transmission line (Z_0 = 50Ω, Td = 1ns) are shown in Fig. 5. In this test the supply voltage of the buffer is kept constant at vdd = 0.7VDDQ in order to highlight the ability of the models to yield accurate results also for supply voltages far from the nominal value. From this comparison, it is clear that the models #2 and #3 provide better results and that model #3 proposed in this paper offer a remarkable high accuracy level in extreme operating conditions as well.

A second test is defined by the same setup used for the example of Fig. 1. This setup is obtained by connecting the example buffer to a real power delivery network and by using the buffer to drive a transmission line with a 50-bit long pseudo-random binary sequence. The transmission line data are Z_0 = 50 and Td = 0.5 ns, and the power delivery network is modeled by a lumped RL series connection (R = 0.2Ω, L = 10nH). The large value of the inductor L is chosen to cause the large fluctuations of the supply voltage v_{dd} shown in Fig. 1. The responses of model #1 and model #3 are shown in Fig. 6 and confirm the accuracy of the proposed model for simulations involving large supply voltage fluctuations.

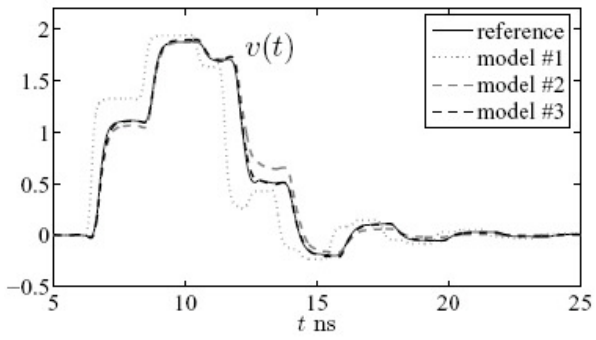

Figure 5: Output port voltage response v(t) of the example buffer that applies the '010' bit stream to an ideal transmission line load while its supply pin is connected to a 70%VDDQ battery (see text for details).

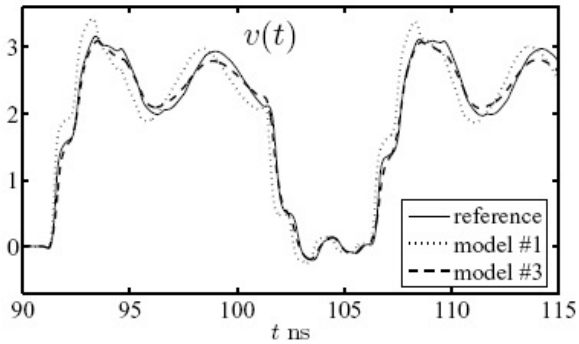

Figure 6: Part of the output port voltage response v(t) of the example buffer loaded by a distibuted interconnect structure. The IC produces a peseudo-random bit stream and its supply pin is connected to a realistic power distribution network (see text for details).

5. Detailed model structure

For the driver of Fig. 1 the selected and complete model representation writes

$$i(t) = w_H(t) \cdot (K_H(v_{dd}) \cdot f_{sH}(v) + f_H(v_{dd} - v, d/dt)) + w_L(t) \cdot (K_L(v_{dd}) \cdot f_{sL}(v) + f_L(v, d/dt) + f_{cL}(v_{dd}, d/dt)) \quad (4)$$

$$i_{dd}(t) = w_H(t) \cdot (K_H(v_{dd}) \cdot f_{sH}(v) + f_H(v_{dd} - v, d/dt)) + w_{dH}(t) \cdot f_{dH}(v_{dd}, v, d/dt) + w_{dL}(t) \cdot f_{dL}(v_{dd}, v, d/dt) + \delta_i(t) \quad (5)$$

Where (to summarize)

- $f_{Sh}(v)$ and $f_{Sl}(v)$ are the i−v output port static characteristics of the driver kept in fixed high and low logic state respectively (the static curves are defined in a tabular format);
- The terms $K_H(v_{dd})$ and $K_L(v_{dd})$ (defined in a tabular format in this version) are suitable coefficients introduced to account for the effects of the large fluctuations of the power supply voltages on the static characteristics [1,2] .
- $w_H(t)$, $w_L(t)$, $w_{dH}(t)$, $w_{dL}(t)$ are time-varying signals accounting for the logic state evolution of the buffer (*i.e.*, they play the same role of the signal v_g);

  All these time-varying signals are composed by means of a suitable concatenation of basic "up" and "down" signals computed during the model estimation process. As an example, $w_H(t)$ is obtained by concatenating the basic sequences $w^{(u)}{}_H(t)$ and $w^{(d)}{}_H(t)$ defined by

$$w^{(u)}{}_H(t) = w^{(u)}{}_H(t + \tau^{(u)}(v_{dd}))$$
$$w^{(d)}{}_H(t) = w^{(d)}{}_H(t + \tau^{(d)}(v_{dd})) \quad (6)$$

  where $\tau^{(u)}(v_{dd})$ and $\tau^{(d)}(v_{dd})$ (defined in a tabular format in this version) are suitable delays introduced to account for the large fluctuations of the power supply voltages [2] and $w^{(u)}{}_H(t)$ and $w^{(d)}{}_H(t)$ are the basic sequences accounting for the up and the down state transitions of the port current $i(t)$ computed for the nominal power supply. The other terms $w_L(t)$, $w_{dH}(t)$, $w_{dL}(t)$ are obtained in a similar way.

- $\delta_i(t)$ is a time-varying signal accounting for the power supply current drawn by the pre-driver stages;
- f_H, f_L, f_{cL}, f_{dH} and f_{dL} are nonlinear input-output parametric models accounting for the nonlinear dynamical effects of the buffer behavior. In this version, such models are discrete-time Local-Linear State-Space (LLSS) representation [3].

  A general LLSS representation describing a MIMO system writes

$$\boldsymbol{x}(k) = \sum_{j=1}^{p} \rho_j\left(\boldsymbol{s}(k-1)\right)\left(\mathbf{A}_j(k-1)\boldsymbol{x}(k-1) + \mathbf{B}_j(k-1)\boldsymbol{u}(k-1) + \boldsymbol{q}_j\right)$$
$$\boldsymbol{y}(k) = \sum_{j=1}^{p} \rho_j\left(\boldsymbol{s}(k-1)\right)\left(\mathbf{C}_j(k)\boldsymbol{x}(k) + \mathbf{D}_j(k)\boldsymbol{u}(k) + \boldsymbol{z}_j\right) \quad (7)$$

$$\rho_j(\boldsymbol{s}(k)) = \frac{\varphi_j(\boldsymbol{s}(k))}{\sum_{i=1}^{p} \varphi_i(\boldsymbol{s}(k))}$$

$$\varphi_j(\mathbf{s}(k)) = \exp(-\|\mathbf{s}(k) - \mathbf{t}_j\|^2 / \beta j^2)$$

$$\boldsymbol{s}(k) = [\boldsymbol{u}^T(k), \boldsymbol{u}^T(k \quad 1), \ldots, \boldsymbol{u}^T(k \quad r)]^T$$

Where

– p is the number of local-linear models
– $\boldsymbol{u} = [\, u_1, \ldots, u_m \,]^T$ is the input vector (size m)
– $\boldsymbol{y} = [\, y_1, \ldots, y_l \,]^T$ is the output vector (size l)
– $\boldsymbol{x} = [\, x_1, \ldots, x_n \,]^T$ is the state vector (size n)
– r is the dynamic order
– $\boldsymbol{s}$ is the scheduling vector (size $m \bullet r$)

Each j-th local linear model is defined by

– the state matrix $\mathbf{A}_j$ (size n x n)
– the input-state matrix $\mathbf{B}_j$ (size n x m)
– the state bias vector $\mathbf{q}_j$ (size n)
– the state-output matrix $\mathbf{C}_j$ (size l x n)
– the input-output matrix $\mathbf{D}_j$ (size l x m)
– the output bias vector $\mathbf{z}_j$ (size l)
– the center vector $\mathbf{t}_j$, size ($m \bullet r$)
– the spreading parameter (β_j)

Tables I and II collect the Matlab variables that will be used to generate a preliminary version of the extended Mpilog tool that implements this extended model structure.

Table I. Matlab variables defining the general structure of the model.

Element	Type	MOD field
f_H	LLSS model (see Tab. II)	NET_H
f_L	LLSS model (see Tab. II)	NET_L
f_{Cl}	LLSS model (see Tab. II)	NET_Cl
f_{Dh}	LLSS model (see Tab. II)	NET_ddH
f_{Dl}	LLSS model (see Tab. II)	NET_ddL
f_{Sh}	i–v table	dcH.i,dcH.v
f_{Sl}	i–v table	dcL.i,dcL.v
K_H	table of coefficients	dcH.coeff,dcH.vdd
K_L	table of coefficients	dcL.coeff,dcL.vdd
$\tau^{(u)}$	table of delays[1]	tdelay4wcoeff.deltat.up, tdelay4wcoeff.deltat.vdd
$\tau^{(d)}$	table of delays[1]	tdelay4wcoeff.deltat.dw, tdelay4wcoeff.deltat.vdd
$w^{(u)}{}_H$	vector[1]	wHu
$w^{(d)}{}_H$	vector[1]	wHd
$w^{(u)}{}_L$	vector[1]	wLu
$w^{(d)}{}_L$	vector[1]	wLd
$\delta^{(u)}{}_i$	vector[1]	Diu
$\delta^{(d)}{}_i$	vector[1]	Did
$w^{(u)}{}_{Dh}$	vector[1]	wdHu
$w^{(d)}{}_{Dh}$	vector[1]	wdHd
$w^{(u)}{}_{Dl}$	vector[1]	wdLu
$w^{(d)}{}_{Dl}$	vector[1]	wdLd

Table II. Matlab variables defining the LLSS submodels.

Element	Type	NET field
p	scalar	p
r	scalar	rsched
m	scalar	m
n	scalar	n
l	scalar	no
$\mathbf{A}_j$	matrix	A{j}
$\mathbf{B}_j$	matrix	B{j}
$\boldsymbol{q}_j$	vector	Q{j}
$\mathbf{C}_j$	matrix	C{j}
$\mathbf{D}_j$	matrix	D{j}
$\boldsymbol{z}_j$	vector	Z{j}
$\boldsymbol{t}_j$	vector	cnt(j,:)'
$1/\beta_j$	scalar	gamma(j)

6. Parameter estimation

This section outlines the procedure for the estimation of model parameter from either simulation or measurement. Without loss of generality, the procedure is detailed for the first part of the model equation defined by (5), i.e. the part related to the output port current, since the estimation of the parameters for the alternate structure (6) follows the same procedure.

[1] time axis is not provided, since all the vectors consist of sequences of uniformly distributed samples (the sampling period is equal to MOD.T).

It is worth noting that the model equation defining (5) is close to the basic structure of the classical model. The only differences rely on the inclusion of the additional coefficients k_H and k_L and of the delays $\tau^{(u,d)}(v_{dd})$ within the definition of the weighting functions w_H and w_L in (7). In basic mpilog model, the $k_{H,L}$ coefficients are equal to 1 and the time delays within $w_{H,L}$ are 0.

Howing to this, the first step of the procedure for the model parameter estimation amounts to apply the state-of-the-art technique developed for the standard Mpilog models. i.e.,

- For the estimation of models from simulation: apply the modeling procedure of [8];
- For the estimation of models form real measured data: apply the procedure of [9]

Then, once the complete set of parameters of the basic model are computed:

- For the estimation of models from simulation: compute the table of coefficients defining $k_H(v_{dd})$ and $k_L(v_{dd})$ from the a set of device i-v port current responses obtained by varying the power supply voltage, as suggested in Sec. 2. Generate the table of delays in (7) by computing the different delays of the up and down state transition events of the output port voltage while the buffer is loaded by a, e.g., 50Ω, resistor. The delays are referred to the switching events of the output port voltage computed for nominal power supply value.
- For the estimation of models from real measured data: use approximate analytical expressions for both the $k_{H,L}$ coefficients and the delays. $k_{H,L}$ can be effectively computed as suggested in Sec. 2 where the complete set of static characteristics are approximated by the analytical MOS equations. On the other hand, $\tau^{(u,d)}(v_{dd}) = \tau_p^{(u,d)} * (1\text{-}VDD/vdd)$; where VDD is the nominal power supply voltage and $\tau_p^{(u,d)}$ is the delay of the up and the down state transition events computed between the input and the output port voltage signals of a buffer circuit for the nominal supply value.

7. Model implementation

Within this activity, the partner involved in the project agree on two different implementations of the model equations to be used in commercial EDA simulation environments. The selected implementation formats are ELDO (a SPICE-based circuit simulation toll of Mentor) and Verilog-A (a very general hardware description language that will be used for the validation of the complete simulation platform based on the Cadence tools).

As an example, the Verilog-A implementation of the model equations for an example device follows.

```
// Output + supply ports driver macromodel (nonlinear static + nonlinear dynamic) (LLSS), 10-Sep-2008
// VERILOG implementation autogenerated by Mpilog
// Minimum Bit Time requirement for accurate results: 6.9825ns

`include "constants.vams"
`include "disciplines.vams"

module drvmod(v,vdd,ref,w1,w2,w3,w4,di,ref2);

        electrical v,vdd,ref,f1,f2,f3,f4,f6,ref2,myVo;
        parameter real VDD=2.5;

        analog
            V(myVo) <+ V(v,ref)-V(vdd,ref)+VDD;

        drvmod_core DRVCORE(v,vdd,ref,w1,w2,w3,w4,di,f1,f2,f3,f4,f6);
        drvmod_fH DRV1(f1,myVo,ref2);
        drvmod_fL DRV2(f2,v,ref2);
```

```
        drvmod_fdH DRV3(f3,vdd,v,ref2);
        drvmod_fdL DRV4(f4,vdd,v,ref2);
        drvmod_fcL DRV6(f6,vdd,ref2);

endmodule

module drvmod_core(v,vdd,ref,w1,w2,w3,w4,di,f1,f2,f3,f4,f6);

    electrical v,vdd,ref,w1,w2,w3,w4,di,f1,f2,f3,f4,f6;
    real fk1,fk2,fs1,fs2;
    parameter real VDD=2.5;

    analog begin

           fs1 = $table_model(V(v,ref)-V(vdd,ref)+VDD,"drvmod_dcH.table","L");
           fs2 = $table_model(V(v,ref),"drvmod_dcL.table","L");

           fk1 = $table_model(V(vdd,ref),"drvmod_coeffH.table","L");
           fk2 = $table_model(V(vdd,ref),"drvmod_coeffL.table","L");

           I(vdd,v)   <+ V(w1)*(fk1*fs1+V(f1));
           I(ref,v)   <+ V(w2)*(fk2*fs2+V(f2)+V(f6));
           I(vdd,ref) <+ V(w3)*(V(f3)) + V(w4)*(V(f4)) + V(di);

    end

endmodule



// drvmod_fH LLSS subcircuit...
module drvmod_fH(f1,u1,xref);

    electrical f1,u1,xref;
    electrical fx1,fx2,fx3,fx4;
    electrical x1,x2,x3,x4;
    parameter real ts=1.75000000000000e-011;

    capacitor #(.c(ts)) C1(x1,xref);
    resistor  #(.r(1))  R1(x1,xref);

    capacitor #(.c(ts)) C2(x2,xref);
    resistor  #(.r(1))  R2(x2,xref);

    capacitor #(.c(ts)) C3(x3,xref);
    resistor  #(.r(1))  R3(x3,xref);

    capacitor #(.c(ts)) C4(x4,xref);
    resistor  #(.r(1))  R4(x4,xref);

    analog begin

              I(xref,x1) <+ V(fx1);
              I(xref,x2) <+ V(fx2);
              I(xref,x3) <+ V(fx3);
              I(xref,x4) <+ V(fx4);

              V(fx1)   <+   (4.12820019648073e-004)+(9.51963560276389e-001)*V(x1)+(3.20567279561518e-
001)*V(x2)+(3.06338606276907e-001)*V(x3)+(-2.77458618052240e-002)*V(x4)+(-7.95321749508382e-
002)*V(u1);

              V(fx2)  <+  (-1.07573962706580e-003)+(-2.12762307302841e-001)*V(x1)+(9.32589979192989e-
001)*V(x2)+(-2.69486932558118e-001)*V(x3)+(-9.27797654104023e-002)*V(x4)+(5.23432961110165e-
002)*V(u1);

              V(fx3)   <+   (-3.71672847765052e-004)+(3.00172600257281e-002)*V(x1)+(8.48242109174080e-
002)*V(x2)+(3.04878483250900e-001)*V(x3)+(-8.40845295067129e-001)*V(x4)+(2.04633221596928e-
001)*V(u1);

              V(fx4)   <+   (-1.71518950901754e-004)+(8.49441872254933e-003)*V(x1)+(2.75707196149580e-
002)*V(x2)+(-2.43573442309752e-001)*V(x3)+(6.68137566732976e-001)*V(x4)+(7.44850670525780e-
002)*V(u1);
```

```
            V(f1)   <+   (-6.92938086866514e-004)+(-3.37291835975618e-001)*V(x1)+(2.04701677831941e-
001)*V(x2)+(8.61437866936576e-001)*V(x3)+(3.12519751241447e-001)*V(x4)+(-2.32496657115571e-
001)*V(u1);
    end

endmodule



// drvmod_fL LLSS subcircuit...
module drvmod_fL(f1,u1,xref);

    electrical f1,u1,xref;
    electrical fx1,fx2,fx3;
    electrical x1,x2,x3;
    parameter real ts=1.75000000000000e-011;

    capacitor #(.c(ts)) C1(x1,xref);
    resistor  #(.r(1))  R1(x1,xref);

    capacitor #(.c(ts)) C2(x2,xref);
    resistor  #(.r(1))  R2(x2,xref);

    capacitor #(.c(ts)) C3(x3,xref);
    resistor  #(.r(1))  R3(x3,xref);

    analog begin

            I(xref,x1) <+ V(fx1);
            I(xref,x2) <+ V(fx2);
            I(xref,x3) <+ V(fx3);

            V(fx1)   <+   (-1.55671254690733e-004)+(5.43165196003401e-001)*V(x1)+(7.95821672741120e-
001)*V(x2)+(-2.78563009590130e-001)*V(x3)+(-1.81375035204585e-001)*V(u1);

            V(fx2)   <+   (1.17150671048076e-004)+(3.10233303720471e-001)*V(x1)+(1.06833886490868e-
001)*V(x2)+(8.26433994059370e-001)*V(x3)+(1.84126906537258e-001)*V(u1);

            V(fx3)   <+   (1.05650103201165e-005)+(-2.41824192786805e-002)*V(x1)+(1.84877706115605e-
001)*V(x2)+(4.88134379700442e-001)*V(x3)+(-3.95249369769007e-002)*V(u1);

            V(f1)   <+   (-1.76041726367731e-004)+(-7.84171316635840e-001)*V(x1)+(5.76052056283368e-
001)*V(x2)+(1.03564001398434e-001)*V(x3)+(-1.93391550057655e-001)*V(u1);
    end

endmodule



// drvmod_fcL LLSS subcircuit...
module drvmod_fcL(f1,u1,xref);

    electrical f1,u1,xref;
    electrical fx1,fx2,fx3;
    electrical x1,x2,x3;
    parameter real ts=1.75000000000000e-011;

    capacitor #(.c(ts)) C1(x1,xref);
    resistor  #(.r(1))  R1(x1,xref);

    capacitor #(.c(ts)) C2(x2,xref);
    resistor  #(.r(1))  R2(x2,xref);

    capacitor #(.c(ts)) C3(x3,xref);
    resistor  #(.r(1))  R3(x3,xref);

    analog begin

            I(xref,x1) <+ V(fx1);
            I(xref,x2) <+ V(fx2);
            I(xref,x3) <+ V(fx3);

            V(fx1)   <+   (-3.16889454456667e-006)+(9.05668087560901e-001)*V(x1)+(4.31762185376446e-
001)*V(x2)+(8.00945622405630e-002)*V(x3)+(-6.38283874270411e-001)*V(u1);
```

```
            V(fx2)   <+   (-4.30926641846462e-006)+(3.78475105073927e-002)*V(x1)+(1.27642108829854e-
001)*V(x2)+(4.31056285086977e-002)*V(x3)+(-8.90071037506425e-001)*V(u1);

            V(fx3)   <+   (-6.76214206347483e-006)+(-7.11707759601821e-003)*V(x1)+(4.05643645601587e-
002)*V(x2)+(-7.02776866734604e-001)*V(x3)+(1.76606420036635e+001)*V(u1);

            V(f1)   <+   (-2.56107953548558e-005)+(-4.40394342310010e-001)*V(x1)+(8.89271432505303e-
001)*V(x2)+(2.19313227780998e-002)*V(x3)+(7.63712538725736e-002)*V(u1);
    end

endmodule



// drvmod_fdH LLSS subcircuit...
module drvmod_fdH(f1,u1,u2,xref);

    electrical f1,u1,u2,xref;
    electrical fx1,fx2,fx3;
    electrical x1,x2,x3;
    parameter real ts=1.75000000000000e-011;

    capacitor #(.c(ts)) C1(x1,xref);
    resistor  #(.r(1))  R1(x1,xref);

    capacitor #(.c(ts)) C2(x2,xref);
    resistor  #(.r(1))  R2(x2,xref);

    capacitor #(.c(ts)) C3(x3,xref);
    resistor  #(.r(1))  R3(x3,xref);

    analog begin

            I(xref,x1) <+ V(fx1);
            I(xref,x2) <+ V(fx2);
            I(xref,x3) <+ V(fx3);

            V(fx1)   <+   (1.03340407484852e-003)+(7.47665837638994e-001)*V(x1)+(6.71834995349459e-
001)*V(x2)+(-5.25864971713034e-001)*V(x3)+(-2.07910395691882e+002)*V(u1)+(-
1.22358350343039e+001)*V(u2);

            V(fx2)   <+   (-3.31843576458792e-004)+(5.93192479345641e-002)*V(x1)+(3.05357424758343e-
001)*V(x2)+(-1.29219466485360e-001)*V(x3)+(-3.18800823250871e+002)*V(u1)+(-
1.87574756420108e+001)*V(u2);

            V(fx3)   <+   (-1.64461149159552e-004)+(1.78941948321628e-003)*V(x1)+(-9.64635750423258e-
003)*V(x2)+(7.48876510534231e-002)*V(x3)+(-6.74710027109640e+002)*V(u1)+(-
3.97001542499715e+001)*V(u2);

            V(f1)   <+   (-9.18105909929486e-004)+(-6.94390331570794e-001)*V(x1)+(6.92314497442010e-
001)*V(x2)+(-1.10094828699250e-001)*V(x3)+(2.15326348737061e+000)*V(u1)+(1.20826817155180e-
001)*V(u2);
    end

endmodule



// drvmod_fdL LLSS subcircuit...
module drvmod_fdL(f1,u1,u2,xref);

    electrical f1,u1,u2,xref;
    electrical fx1,fx2,fx3;
    electrical x1,x2,x3;
    parameter real ts=1.75000000000000e-011;

    capacitor #(.c(ts)) C1(x1,xref);
    resistor  #(.r(1))  R1(x1,xref);

    capacitor #(.c(ts)) C2(x2,xref);
    resistor  #(.r(1))  R2(x2,xref);

    capacitor #(.c(ts)) C3(x3,xref);
    resistor  #(.r(1))  R3(x3,xref);
```

```
    analog begin

              I(xref,x1) <+ V(fx1);
              I(xref,x2) <+ V(fx2);
              I(xref,x3) <+ V(fx3);

              V(fx1)    <+    (5.81647772203841e-004)+(7.38710797423065e-001)*V(x1)+(7.55751423606151e-
001)*V(x2)+(-6.58092971621826e-003)*V(x3)+(1.18220271052213e+000)*V(u1)+(-2.49615406082889e-
002)*V(u2);

              V(fx2)    <+    (5.12031511780797e-005)+(4.52444250404271e-002)*V(x1)+(5.93237626699601e-
001)*V(x2)+(-4.92024263966865e-001)*V(x3)+(-9.46708818230355e-001)*V(u1)+(1.98023151114946e-
002)*V(u2);

              V(fx3)    <+    (5.63818919704159e-004)+(1.68720947538740e-002)*V(x1)+(-3.23148043451237e-
001)*V(x2)+(4.63193416161257e-001)*V(x3)+(-8.57574705504259e-001)*V(u1)+(1.79079652718963e-
002)*V(u2);

              V(f1)    <+    (-2.12652761904219e-005)+(-6.94332891112927e-001)*V(x1)+(6.71899281616148e-
001)*V(x2)+(1.17453534355607e-001)*V(x3)+(1.87734207123490e+000)*V(u1)+(-3.97945294437581e-
002)*V(u2);
    end

endmodule
```

It is worth to remark that some parameters included in Table I and used by the above implementation are stored in the following external ASCII data files:

- drvmod_coeffH.table: table of coefficients (see the element K_H in the model equations and in Tab. I)
- drvmod_coeffL.table: table of coefficients (see the element K_L in the model equations and in Tab. I)
- drvmod_dcH.table: i-v static characteristic of the buffer in high state (see the element f_{Sh} in the model equations and in Tab. I)
- drvmod_dcL.table: i-v static characteristic of the buffer in high state (see the element f_{Sl} in the model equations and in Tab. I)
- drvmod_delaydw.table: table of delays (see the element $\tau^{(u)}$ in the model equations and in Tab. I)
- drvmod_delayup.table: table of delays (see the element $\tau^{(d)}$ in the model equations and in Tab. I)

As an example, the file “drvmod_delayup.table” writes

```
1.7500000e+000 -4.6587729e-002
2.5000000e+000  0.0000000e+000
3.2500000e+000  6.0322064e-003
```

8. Model validation

In the activities of the next tasks of WP2 (Tasks 2.2 and 2.3), the proposed model structure will be used to generate the models of a set of example drivers selected by Numonyx. The reference models will be either transistor-level description of buffers and real devices, thus leading to the estimation and validation of models from either simulation or measurements. In this framework, the assessment of accuracy and efficiency of the proposed models will be done by comparing the responses of the two implementations of the models and the reference simulated or measured responses while the devices are connected to different interconnect structures and operates in normal condition.

9. General Overview of the Proposed Model

This Section describes a top-down strategy that, starting with a general I/O buffer model structure, uses physical knowledge of the typical I/O buffer circuit to prune such general model, ending on a simplified structure. It is shown that the proposed Mπlog model is representative of such final structure.

Looking at the general I/O buffer circuit from a modeling perspective, such system can be thought as a four single-port nonlinear network as shown in Fig. 7. Each port has associated a certain voltage (referred to a common reference, not show in the figure), and the respective input current.

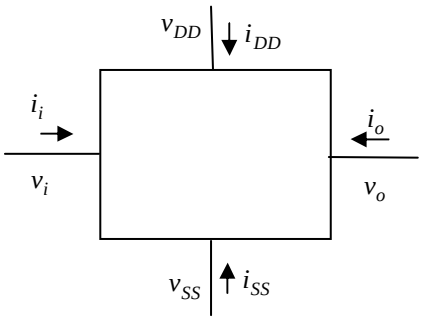

Fig. 7 – General model of the I/O buffer circuit.

Since it is admissible that in a real I/O buffer circuit the power supply voltages (v_{DD} and v_{SS}) may vary with time, these are seen as variables to be considered in the model. In fact, although the presented voltages (input voltage v_i, output voltage v_o, and v_{DD} and v_{SS} for the power supply) are all relative to an external reference voltage, the I/O buffer circuit is only sensitive to voltage differences between its terminals. In other words, in spite of existing four voltage variables, only three of them are independent variables because it is their relative difference that matters for the I/O buffer circuit. Thus, one of them can be selected to be the reference voltage from the I/O buffer perspective, and the choice that seems more natural from the circuit level analysis is the v_{SS} voltage. This, however, does not compromise the model generality, namely in its capability to represent v_{SS} rail voltage fluctuations.

Moreover, and assuming that the IC core behaves like an almost ideal voltage source when generating the v_i voltage, then the input current i_i will not be a matter to be considered since the v_i voltage is sufficient to the characterization of the input single-port. Notice that, even though there might be currents charging and discharging the gate capacitances (C_{gs}) of the MOSFETs composing the I/O buffer, this inner current can still be considered as an internal part of the model, but it is irrelevant (under this assumption) to consider that such current has its origin outside the I/O buffer.

Under the above description, a set of general equations can be formed to describe the interdependence of the model variables. This is shown in (8).

$$\begin{cases} i_i = 0 \\ i_{SS} = -\left(i_i + i_o + i_{DD}\right) \\ i_{DD} = f_{DD}\left(v_i - v_{SS}, v_{DD} - v_{SS}, v_o - v_{SS}, i_i, i_o, i_{DD}\right) \\ i_o = f_o\left(v_i - v_{SS}, v_{DD} - v_{SS}, v_o - v_{SS}, i_i, i_o, i_{DD}\right) \end{cases} \tag{8}$$

As for the voltages only the three voltage differences are independent variables, the charge conservation principle implies that the sum of all currents entering the I/O buffer is equal to zero [second equation of (8)], which means that only three currents are independent variables. Moreover, with the assumption that the input current i_i is null, only two currents remain as independent variables.

The inclusion of the dynamic behavior of the I/O buffer can be considered by expanding the number of independent variables to the past samples of the voltage differences and currents. Equations (9) illustrate this expansion.

$$\begin{cases} \bar{v}_i = v_i - v_{SS} \\ \bar{v}_{DD} = v_{DD} - v_{SS} \\ \bar{v}_o = v_o - v_{SS} \\ i_{DD}(t) = f_{DD}\begin{bmatrix} \bar{v}_i(t),\cdots,\bar{v}_i(t-M\tau),\bar{v}_{DD}(t),\cdots,\bar{v}_{DD}(t-M\tau),\bar{v}_o(t),\cdots,\bar{v}_o(t-M\tau), \\ i_o(t),\cdots,i_o(t-M\tau),i_{DD}(t),\cdots,i_{DD}(t-M\tau) \end{bmatrix} \\ i_o(t) = f_o\begin{bmatrix} \bar{v}_i(t),\cdots,\bar{v}_i(t-M\tau),\bar{v}_{DD}(t),\cdots,\bar{v}_{DD}(t-M\tau),\bar{v}_o(t),\cdots,\bar{v}_o(t-M\tau), \\ i_o(t),\cdots,i_o(t-M\tau),i_{DD}(t),\cdots,i_{DD}(t-M\tau) \end{bmatrix} \end{cases} \quad (9)$$

where M is the memory span considered for the system [notice that, in the case of equation (9) the memory span was considered to be equal for all variables – in a more general case, different spans might be considered] and τ is the time interval between consecutive samples. The functions $f_{DD}(\cdot)$ and $f_o(\cdot)$ are now purely static nonlinear functions, operating on a $5(M+1)$ set of input variables.

The functions $f_{DD}(\cdot)$ and $f_o(\cdot)$ model the nonlinear dynamic behavior of the I/O buffer, and constitute therefore the core of the behavioral model. They can be implemented through different formulations. The most general approach is to consider universal approximators such as Time-Delay Artificial Neural Networks (TDANN), Time-Delay radial Basis Functions (TDRBF) or Volterra series (VS). Unfortunately, this generality must be paid in terms of a huge number of parameters required to model any practical device, and their relation to particular physical aspects of the system is not usually evident, if not impossible to identify.

One important aspect of (9) is the fact that the last two equations form a set expressed in the implicit form, where the current sample value of the dependant variables $i_{DD}(t)$ and $i_o(t)$ depend instantaneously on themselves. This is very demanding during simulation since it requires a considerable computational effort to solve such implicit nonlinear equations. Fortunately, as with many other physical systems, the I/O buffer circuit is a fading memory system (and, evidently, a causal and stable one) and so, supported by nonlinear system identification theory, equations (9) can be approximated by a different function, now expressed in its explicit form (10), to the required accuracy. The explicit form is much more convenient than the implicit one especially for practical implementations issues of the model (simulation algorithms are less demanding and much more time-efficient).

$$\begin{cases} \bar{v}_i = v_i - v_{SS} \\ \bar{v}_{DD} = v_{DD} - v_{SS} \\ \bar{v}_o = v_o - v_{SS} \\ i_{DD}(t) = f'_{DD}\begin{bmatrix} \bar{v}_i(t),\cdots,\bar{v}_i(t-M\tau),\bar{v}_{DD}(t),\cdots,\bar{v}_{DD}(t-M\tau),\bar{v}_o(t),\cdots,\bar{v}_o(t-M\tau), \\ i_o(t-\tau),\cdots,i_o(t-M\tau),i_{DD}(t-\tau),\cdots,i_{DD}(t-M\tau) \end{bmatrix} \\ i_o(t) = f'_o\begin{bmatrix} \bar{v}_i(t),\cdots,\bar{v}_i(t-M\tau),\bar{v}_{DD}(t),\cdots,\bar{v}_{DD}(t-M\tau),\bar{v}_o(t),\cdots,\bar{v}_o(t-M\tau), \\ i_o(t-\tau),\cdots,i_o(t-M\tau),i_{DD}(t-\tau),\cdots,i_{DD}(t-M\tau) \end{bmatrix} \end{cases} \quad (10)$$

Thus, and considering a memory span of M+1 samples, it would be required to obtain two mapping functions from a 5M+3 space into a one-dimensional one.

These two mappings can be further pruned if it is put into evidence more specific knowledge of the physical relationships of the variables in action. For example, after performing several SPICE simulation tests of a typical I/O buffer structure, it was concluded that the main source of nonlinear dynamics of an I/O buffer comes from the recursive interaction between the buffer load (typically, a linear RLC network characterizing the transmission lines from the I/O buffer ends to the respective bonding pads) and the inner behavior of the transistors. This recursive behavior can be easily explained by the dependence that the load voltage has on the drain current produced by the transistors, and such current is dependant on the drain-source voltage which is exactly the load voltage.

Also deduced from the simulation tests, where different load characteristics were tested, was the result that the dominant source of memory (dynamics) within an I/O buffer is, in fact, the linear load whose effect is, again, fed-back to the transistor. The transistor characteristic that is more relevant is the fact that its current-voltage relationship is nonlinear (its internal dynamics is not very relevant when compared to the transistor-load interaction.

As mentioned before, the equations that generate $i_{DD}(t)$ and $i_o(t)$ in (9) can be synthesized by universal approximation functions. Nevertheless, physical reasoning based pruning is desirable so that efficiency can be obtained from the model simulations (and also for the model extraction procedure). For instance, the switching between the output NMOS and PMOS as a function of the input digital voltage produces an effect on the functions $f'_{DD}(\cdot)$ and $f'_o(\cdot)$. The proposed Mπlog model described in the above sections includes this pruning though the weighting functions $w_H(t)$ and $w_L(t)$. Moreover, the observed fact of the dominance of transistor static nonlinearity over transistor inner dynamics is also contemplated within the proposed Mπlog model with the inclusion of the functions $f_{sH}(v)$ and $f_{sL}(v)$ which are the static nonlinear curves of the current-voltage characteristics of the transistors. As shown in (9), the v_{DD} voltage is also an input variable to the current $i_o(t)$ determination and this was incorporated on the Mπlog model by means of weighting functions [$K_H(v_{dd})$ and $K_L(v_{dd})$] on the transistor nonlinear static characteristics. This is a reasonable approach when the observation of the transistor characteristics is considered for varying drain-gate and drain-source voltages. The remaining contribution for the current dependency specification is the dynamics essentially generated by the load-transistor interaction. Once again, a term for such purpose is included in the Mπlog model by means of the functions $f_H(\cdot)$, $f_L(\cdot)$, $f_{cL}(\cdot)$, $f_{dL}(\cdot)$ an $f_{dH}(\cdot)$. These use a general Local Linear State-Space as a basis structure for their implementation which are, as analyzed above, in an explicit form.

Thus, in conclusion, it was shown how the proposed Mπlog model is, in fact, a general I/O buffer model which incorporates physical knowledge of the I/O buffer circuit (and of its inner devices) to generate a pruned and more efficient model.

10. References

[1] I/O Buffer Information Specification (IBIS) Ver. 5.0, on the web at http://www.eigroup.org/ibis/ibis.htm, Aug. 2008.

[2] I. S. Stievano, I. A. Maio, F. G. Canavero, "M[pi]*log* Macromodeling via Parametric Identification of Logic Gates," *IEEE Trans. on Advanced Packaging*, Vol. 27, No. 1, pp. 15–23, Feb. 2004.

[3] I. S. Stievano, I. A. Maio, F. G. Canavero, C. Siviero, "Reliable Eye-Diagram Analysis of Data Links via Device Macromodels," *IEEE Trans. on Advanced Packaging*, Vol. 29, No. 1, pp. 31–38, Feb. 2006.

[4] A. Muranyi, A. Girardi, G. Bernardi, R. Izzi, "Gate Modulation Effect (table format)," BIRD #98.1, On the web at http://www.vhdl.org/pub/ibis/birds/, May 20, 2005.

[5] Arpad Muranyi, "Gate Modulation Effect," BIRD #97.2, On the web at http://www.vhdl.org/pub/ibis/birds/, Mar. 4, 2005.

[6] J. Sjoberg et al., "Nonlinear black-boxmodeling in system identification: a unified overview," *Automatica*, Vol. 31, No. 12, pp. 1691–1724, 1995.

[7] P. Pulici, A. Girardi, G. P. Vanalli, R. Izzi, G. Bernardi, G. Ripamonti, A. G. M. Strollo, G. Campardo, "A Modified IBIS Model Aimed at Signal Integrity Analysis of Systems in Packag,e" IEEE Trans. On Circuits and Systems, Vol. 55, No. 7, Aug. 2008.

[8] I. S. Stievano, I. A. Maio, F. G. Canavero, "Behavioral models of IC output buffers from on-the-fly measurements," IEEE Transactions on Instrumentation and Measurement, vol. 57, No. 4, pp. 850-855, 2008.

[9] I. S. Stievano, C. Siviero, F. G. Canavero, I. A. Maio, "Behavioral modeling of digital devices via composite local-linear state-space relations," IEEE Transactions on Instrumentation and Measurement, Vol. 57, No. 8, pp. 1757-1765, Aug. 2008.