Macromodeling of Digital I/O Buffers

W. Dghais

Abstract — This paper addresses the state of the art of macromodels' development for high speed digital output buffers for Signal Integrity (SI) analysis and simultaneous switching noise (SSN). An overview of the behavioral black-box techniques used by the Input/Output Buffer Information Specification (IBIS) and the previous nonlinear parametric models is presented. Then, the performance of a sigmoid basis functions' artificial neural network is investigated to model the device behavior. The proposed alternative is based on the estimation of suitable mathematical equations reproducing the external behavior of the device via system identification theory.

Index Terms — Digital output buffer/driver, Macromodeling, sigmoid basis function, signal integrity, system identification.

I. INTRODUCTION

The trend in digital electronic systems is directed towards higher operation frequency, complex packaging density and larger I/O ports count. This is due to advances in fabrication technology and creates a completely new design scenario. As a consequence, the accurate prediction of the signals propagating on system interconnects becomes increasingly important for design engineers because it allows to perform both SI analysis and Electromagnetic Compatibility (EMC) assessment. Such prediction is carried out in a time domain circuit simulator to allow the detailed analysis of the interactions between the digital Integrated Circuits (IC) periphery devices and the loading interconnects lines.

Digital Input/Output buffers are the key components to successfully transmit data between electronic devices. They play an important role for SI and SSN simulation and timing analysis because they contain complex functional parts and are in line with the IC's high number of pins. Moreover, I/O buffers drive high currents when compared to IC logic core due to their large transistors size. Thus, they are the bottleneck of the IC in terms of switching speed. For these reasons, macromodeling I/O buffers to capture their nonlinear dynamic behavior are a challenging task that motivates this research activity [1]-[4]. In fact, output buffers (drivers) are more difficult to model, due to their strongly nonlinear and dynamic properties, than the input buffers (receivers) which can be often approximated as linear capacitive and resistive impedance components.

In this paper, we present the previous driver macromodels discussing the main advantages and limitations of each approach to then proceed to an alternative parametric nonlinear dynamic behavioral model based on Sigmoidal Basis Function (SBF) Artificial Neural Networks (ANN. The remaining of this paper is organized as follows. Section II reviews the work that has been developed to model the problem at hand. Section III describes in detail the proposed modeling process using SBF ANN as an example and Section IV gives a brief conclusion to the work.

II. STATE OF THE ART

Basically, macromodels can be classified in two classes: physically based and behaviorally based models. In fact, the traditional method for output buffers modeling amounts to describe the device behavior by means of a detailed physical model based on its internal structure. This transistor-level description, considered as the reference for the device, provides the most accurate simulation result. Unfortunately, the model completely discloses the internal details of the device. In addition, the use of complex output driver transistor level models, which are generally large in size, place a large computation burden on the circuit solver, especially in the case of a large multi-IO analysis such as SSN. Therefore, the simulation is highly time consuming and needs considerable resources in term of CPU computation and memory storage. Moreover, they are non-portable, since they rely on encrypted libraries specific to certain simulators.

One way of reducing this complexity has been to use driver macromodels based on the behavioral observation of the input and output relationship that yields to computationally much simpler models.

In the following subsections we will detail the behavioral macromodels based on an equivalent circuit model (IBIS) and the nonlinear parametric models.

A. IBIS method

There have been several techniques proposed for generating driver behavioral macromodels over the last decade. The most popular of such methods is IBIS [1]. The core of IBIS consists of lookup tables of current versus voltage and voltage-time to describe the pull-up, pull-down, and clamp diodes along with some package information, as shown in Fig. 1.

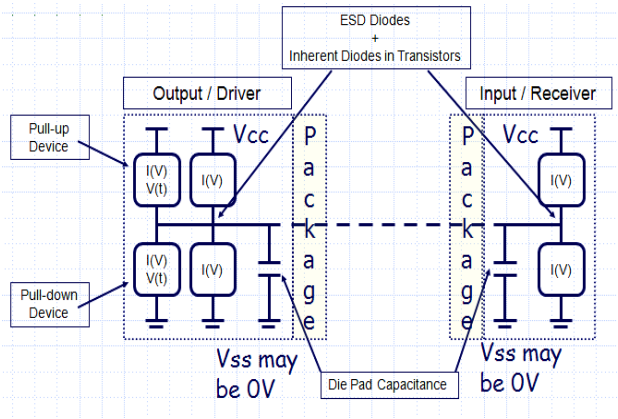

Fig. 1: Key portions of an IBIS driver and receiver model

As driver technology gets increasingly complicated and rise time of input signal gets increasingly smaller, important considerations such as SSN becomes a major consideration when simulating multiple IO drivers in the IC. The enhancement of IBIS model was focused on SSN simulation [2], [3] by addressing its major deficiencies, mainly the predriver and crossbar currents and the local power and ground signals bounce. That has been accommodated in the IBIS model without the need to change its original structure and philosophy. This is done by complimenting the IBIS model with a black-box that contains the error function parameters between the original IBIS and the transistor netlist model. First, for the predriver current error correction, a correlation was observed Fig.2 (b) [2] between the difference current at the V_{DD} pin ($I_{diff,VDD}$) that only occurs during the up and down transition Fig. 2(a) [2] and the voltage levels between the $V_{DD,IBIS}$ and $V_{SS,IBIS}$. This was corrected by estimating polynomial coefficients for voltage controlled current source implementation:

$$I_{diff,VDD} = I_{trans,VDD} - I_{IBIS,VDD} = f\left(V_{DD,IBIS} - V_{SS,IBIS}\right) \quad (1)$$

Besides, the gate modulation effect error was corrected by adding a black box submodel that captures the current difference at the output as a function of the power and ground voltage. $I_{IBIS,out}$ in (2) is obtained using an IBIS model where the output current for the device is extracted from current-voltage, I-V, tables. To account for the power supply variations, a scaling coefficient *K* is introduced (2). *K* is a function of the instantaneous power supply voltage and the voltage at which the I-V tables in the IBIS models were created (2). The effective output current thus scales corresponding to the actual voltage in the power and ground nodes.

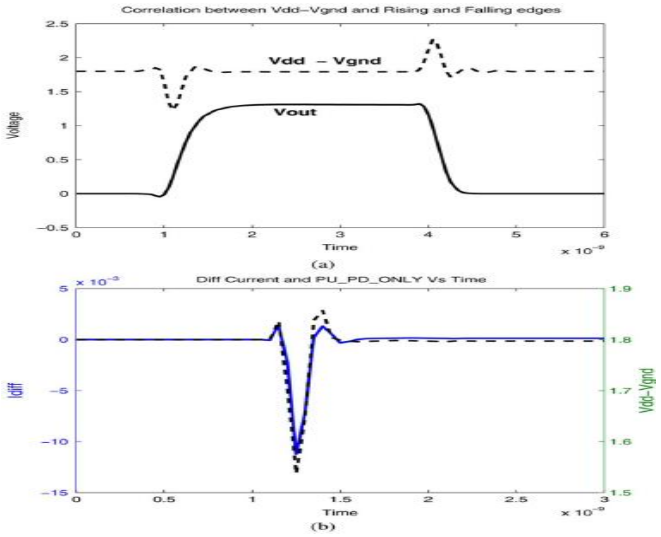

Fig. 2. Power drop and ground bounce correlate with the switching of the buffer (a). There is a high correlation between the difference in current (solid line) at the V_{DD} pin between the transistor level model and the behavioral (IBIS) model and V(Pwr-Gnd) (dashed line) (b).

$$\begin{cases} I_{diff,out} = I_{trans,out} - I_{IBIS,out} = f\left(V_{DD,IBIS} - V_{SS,IBIS}\right) \\ I_{eff} = K * I_{diff,out} \\ K = N * \dfrac{V_{inst}}{V_{nom}} \end{cases} \quad (2)$$

V_{nom} is the normal value at which the I-V tables in the IBIS models are created. V_{inst} is the instantaneous voltage between the power and ground nodes of the I/O buffer. This voltage is not constant as V_{nom}, but reflects the noise due to switching in the local power and ground nodes as shown in Fig. 5(a). *N* is a user dependent factor for adjusting the *K* factor and is usually the number of the drivers that are switching simultaneously in the system.

Despite its commercial success, IBIS model has intrinsic limitations. In fact, the simulation or measurement strategies have been pre-determined by the IBIS specification in order to generate the experimental data supported by EDA tools. This makes IBIS model unsuitable to include the higher-order dynamics presented by cutting-edge driver technologies, as it mainly relies on the static I-V information revealed on the DC characteristics.

It was motivated by these IBIS model deficiencies that, recently, a newer class of parametric black-box methods [4]-[7] was proposed to better capture the I/O driver dynamics, via system identification theory. In addition to the static I-V characteristics used in IBIS models, this class of methods uses Radial Basis Functions (RBF) [??] or Spline Function With Finite Time Difference (SFWTD) [??] in order to obtain "higher-order" accuracy in capturing the output behavior of driver circuits.

B. Radial Basis Functions (RBF) method

The discrete time parametric nonlinear dynamic model representation can be written as:

$$\begin{cases} i_0(k) = F[\theta, x(k)] \\ x(k) = [i_0(k-1)..i_0(k-r), v_0(k)..v_0(k-r)]^T \end{cases} (3)$$

where $F[.]$ is a suitable mathematical representation, depending on the parameters collected in vector Θ and the regressor vector $x(k)$ as shown in Fig. 3.

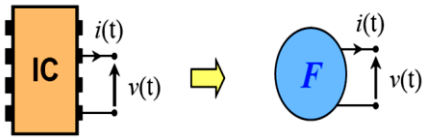

Fig. 3: The observed electrical behavior IC driver circuit is reproduced by a suitable F[.] mathematical relation.

In (3) the output buffer output current is expressed in terms of the output buffer voltage using a summation of basis functions. To create such a model, one needs to carefully stimulate the output port of the buffer to expose its dynamics. The data obtained is then fitted with either RBFs or splines. Finally, the generated macromodel is represented as an equivalent sub-circuit, which is implemented in SPICE and simulated with load interconnects. It has been shown (e.g., [5]-[7]) that these methods are capable of representing the driver circuit quite well and to capture several effects like crosstalk and SSN.

The RBF approach follows the work done by Professor Canavero's group at Politecnico di Torino, in Italy. The output current and voltage are related using a piece-wise parametric formulation:

$$i_0(k) = w_1(k).f_1[\theta_1, x(k)] + w_2(k).f_2[\theta_2, x(k)] \quad (4)$$

$$f_n[\theta_n, x(k)] = \sum_{j=1}^{M} \theta_{nj}\phi\left(|x - c_{nj}|, \beta\right), n = 1,2 \quad (5)$$

In (4), i_0 is the output buffer output current, $f_1[.]$ and $f_2[.]$ are the sub-models that relate the output buffer output current to the output voltage for the buffers input digital HIGH and LOW states, respectively. The transition from one logic state to another is done with the help of weighting functions $w_1(t)$ and $w_2(t)$. These time-varying weighting functions act as switches between the sub-models $f_1[.]$ and $f_2[.]$ which are expressed as a summation of radial basis functions (5), where M is the number of basis functions needed for $f_1[.]$ or $f_2[.]$ to accurately model the digital driver. In (5), Φ is the asymptotically increasing, or decreasing, basis function and θ_j is the weight of the basis function Φ. The centers of the basis functions are defined by c_j and the width, or spread, parameter is defined by β. The *regressor vector x* in (6) collects the past r samples of the output buffer output voltage (v_0) and the output buffer output current (i_0) along with the present sample of the driver output voltage. The parameter r has been called the dynamic order of the model. The dynamic order adopted for the model depends on the complexity of the output buffer that is being modeled.

$$x(k) = [i_0(k-1)\ldots., i_0(k-r), v_0(k)\ldots.v_0(k-r)]^T \quad (6)$$

Although RBF models approximate driver circuits accurately and have been applied to complex circuits with multiple ports [8]-[10], they have some inherent limitations. One of such limitations is the rise of model complexity with the decrease in input rise time for the driver circuit because the driver dynamic characteristics start dominating the static characteristics. This leads to a significant increase in the number of basis functions needed to accurately model the driver circuit. Moreover, the efficiency of black-box methods relies heavily on the choice of model representations, data sets generation and interpretation [7].

C. Spline Function With Finite Time Difference (SFWTD) method

The SFWTD method basically follows the research work carried out by Mutnury and his co-authors in Georgia Institute of Technology, in the USA. This method takes into account both the static and the dynamic characteristics of the drivers such as the current–voltage relationship and memory effects.

The output current can be expressed as a function of the output voltage using static characteristics. Let submodels $f_{1,s}$ and $f_{2,s}$ represent the static characteristic relation between the driver output current and output voltage for a driver when the driver input is set HIGH and LOW, respectively:

$$f_{n,s}(k) = A_{n,m}.v_0^m(k) + A_{n,m-1}.v_0^{m-1}(k) + \cdots$$

$$\ldots + A_{n,0} n = 1,2; m \geq 1 \quad (7)$$

where, A_s are constants, v_0 is the driver output voltage, and the polynomial degree m is usually less than 5. When the driver input is set HIGH at time instant *(k -1)*, the current at the output, i_{oh}, can be expressed as :

$$f_{1,s}(k) = i_{oh}(k-1) = A_{1,m}.v_0^m(k) + A_{1,m-1}.v_0^{m-1}(k) + \ldots$$

$$\ldots + A_{1,0} \quad (8)$$

The incremental change in the driver output current Δi_{oh} is the difference between the present instance *(k)* and previous time instance *(k – 1)* values of sub-model $f_{1,s}$ as shown:

$$f_{1,s}(k) - f_{1,s}(k-1) = i_{oh}(k) - i_{oh}(k-1) = \Delta i_{oh} \quad (9)$$

or

$$f_{1,s}(t) - f_{1,s}(t-\Delta t) = \Delta i_{oh} \quad (10)$$

Once Δi_{oh} is calculated, the first derivative of driver output current i'_{oh} can be approximated as:

$$\frac{f_{1,s}(t) - f_{1,s}(t-\Delta t)}{\Delta t} = \frac{\Delta i_{oh}}{\Delta t} \quad (11)$$

Where Δt is the sampling time. The effect of dynamic behavior when the driver input is HIGH is captured in i'_{oh}. Similarly, the effect of dynamic behavior when the driver input is LOW is captured by i'_{oh}. Therefore, dynamic behavior can be added to static submodels f_1 and f_2 as shown in (4)

$$\begin{cases} f_1(k) = f_{1,s}(k) + p * i'_{oh} \\ f_2(k) = f_{2,s}(k) + pp * i'_{ol} \end{cases} \quad (12)$$

where *p* and *pp* are constants whose magnitude can be estimated by calculating the least mean square error between $f_{1,s} | f_{2,s}$ and the transistor-level driver output current values for inputs High/Low, respectively. It is important to note that there is no limitation on the number of previous output current time instants that can be added to the static sub-models:

$$\begin{cases} f_1(k) = f_{1,s}(k) + p * i'_{oh} + q * i'_{oh} + \cdots \\ f_2(k) = f_{2,s}(k) + pp * i'_{ol} + qq * i'_{ol} + \cdots \end{cases} \quad (13)$$

Once f_1 and f_2 are estimated for input HIGH and LOW, the relationship between the driver output current and voltage can be expressed as shown in (4).

Typically, for most of the driver circuits, SFWFTD models need one previous time instance to accurately model the driver current voltage characteristics. Unfortunately, as the model is based on a polynomial expansion, it has inherent local approximating properties. On the other hand, its parameters can be extracted very easily using the least squares regression method. In fact, we cannot evaluate the polynomial model outside the input range where its parameters were extracted because the error will increase dramatically [10].

III. SIGMOID BASIS FUNCTION BASED ARTIFICIAL NEURAL NETWORK PARAMETRIC MODELING PROCESS

As we could see from the discussion of the poor extrapolation properties of polynomials, it would be better to use models based on activation functions, or basis functions, which are bounded in the output amplitude [10] (e.g. sigmoidal functions Fig. 4). The feed-forward ANN presents a good alternative to model the static characteristics with less parameters because the ANN constitute global approximants in modeling strongly nonlinear systems, do not share the catastrophic degradation of polynomials outside the zone of training and its basis functions show an input output behavior that is similar to the one expected for the I-V relationship of the pull-up and pull-down devices.

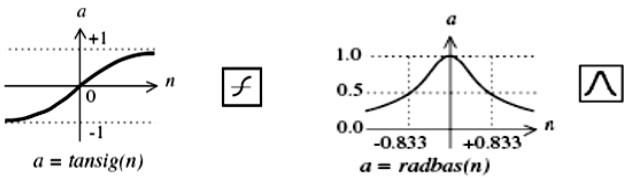

Fig. 4: Tan-Sigmoid and Radial Basis activation functions used in ANN are bounded in output amplitude.

In the following subsections we will illustrate the parametric model extraction procedure by means of the SBF alternative to model the driver static nonlinearity and its mild dynamics along with the power and ground bounce analysis.

A. SBF parametric model selection

For any kind of technology, the driver circuits are composed by a cascade of inverter stages with growing driven capabilities interfacing the internal logic core and loading the external package components. The generic driver structure is depicted in Fig. 5. The final stage's transistors of the drivers are arranged as a pull-up and pull-down network, which contribute to the dominant electrical behavior of the device (high dimensions of the MOSFETS), while the pre-drivers mostly act as a resistive path feeding the Cgs capacitances of the last stage.

The time varying nature of the device due to switching propriety implies the use two separate SBF submodels accounting for both the static and dynamic effect:

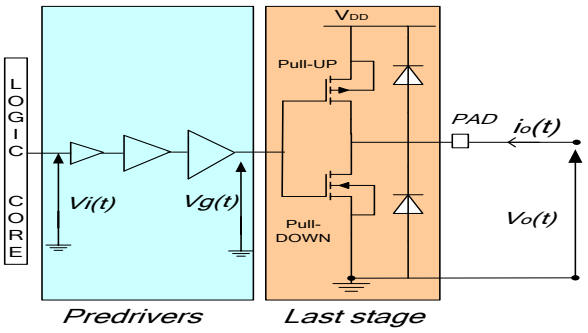

Fig. 5: Generic multistage driver and its relevant electric variables.

$$\begin{cases} i_0(k) = w_1(k).f_1[\Theta_1, x(k)] + w_2(k).f_2[\Theta_2, x(k)] \\ x(k) = [i_0(k-1)..,i_0(k-r), v_0(k)..v_0(k-r)]^T \end{cases} \quad (14)$$

$$\begin{cases} f_n = C + \sum_{i=1}^{M} B_i.g(W_i.v_0 + A_i) \;\; n = 1,2 \\ g(x) = \dfrac{e^x + e^{-x}}{e^x - e^{-x}} \end{cases} \quad (15)$$

Where M is the number of sigmoid activation functions needed to accurately mimic the nonlinearity. Since the model is nonlinear with respect to its parameters, the Levenberg-Marquart nonlinear optimization algorithm is used to estimate the A_i, B_i, C and W_i parameters, minimizing the mean square error cost function between the model and the transistor level current outputs [5]. It is worth to note that both IBIS and the previous nonlinear parametric modeling approaches use the two-piece model representation.

B. Nonlinear static behavior model

Since the nonlinear static characteristic contribute to the dominant behaviour of the driver, this section presents an alternative to describe this nonlinearity using SBF expansions [11]. Besides, the nonlinear I-V characteristic of last stage driver transistors behave as hyperbolic tangents. That is why this model structure is believed to be more suitable for fitting the actual constitutive relations of IC drivers and usually lead to simpler macromodels than those based on RBF. The SBF model of the static characteristic is the following:

$$i_0(k)) = w_1(k).f_1[(\Theta_1, v_0(k))] + w_2(k).f_2[\Theta_2, v_0(k)] \quad (16)$$

The model representation contains no delays (previous or past sample) for either the input or the output signals, nor it depends on the output. Then, the estimation of parameters (weighting and bias) of the static representation sub-model $f_1[.]$ and $f_2[.]$ can be performed following a simple feed-forward topology. This ANN model has one or more hidden layers of tanh-sigmoid neurons followed by an output layer of linear neurons as shown in Fig. 6. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range -1 to +1.

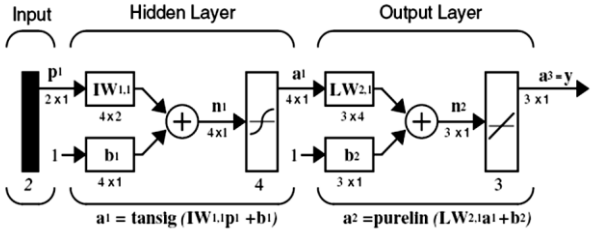

Fig. 6: Structure of feed-forward ANN (source: MATLAB).

In addition, feed-forward ANN is nonlinear in these parameters, which requires a nonlinear parameter identification process, or optimization algorithm, for adjusting the weights and biases. This imposes some limitations such as the need for providing an initial condition for the parameter vector, that the algorithm may not converge and the problem of converging to one of a potential large set of local minima.

The estimation of sub-model parameters of f_1 and f_2 requires the identification signals (v_0 and i_0) which are obtained by applying a triangle voltage waveform to the output terminals and by recording the corresponding output current. Although this could be done at the laboratory, it is usually performed with a transient response of the transistor-level simulation from cadence using spectre simulator, while the buffer input is in a fixed logic state. Such an experiment is described by the ideal setup of Fig. 7.

The training starts by loading the I-V data and adjusting the number of hyperbolic tangent hidden neurons. Then, we divide the set of data into training and validation data to improve the generalization properties of the network. After that, we should scale the input-output data between -1 and +1. After training the network using the optimization Levenberg-Marquardt algorithm, we should be able to extract the parameters that fit the driver submodels nonlinear static characteristic.

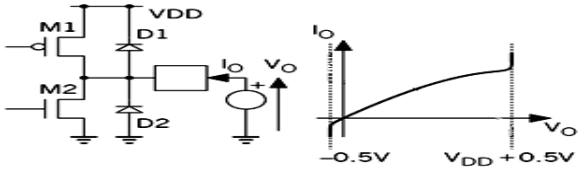

Fig. 7: Simulation setup for the steady state identification signals for submodels. The clamp diodes start to conduct as the output voltage exceeds the supply voltage bounds.

During the training process, the weight and bias parameters are iteratively adjusted to minimize the mean square error cost function:

$$\Theta | \min \left\{ \frac{1}{N} \sum_{k=1}^{N} \left(\overline{i_o}(k) - i_o(k) \right)^2 \right\} \quad (17)$$

where $\overline{i_o}(k) = \overline{i_o}(k.T_s)$ is the sampled output identification signal, T_s is the sampling period, N is the total number of samples and $i_o(k)$ is the response of model (4) to the sampled input identification signal $\overline{v_o}(k)$, as illustrated in Fig. 8.

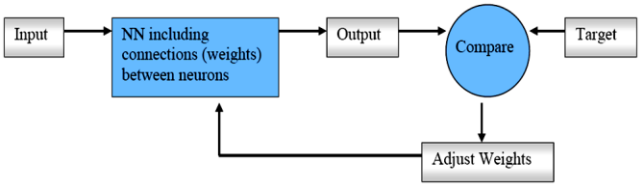

Fig. 8: The weight and bias parameters are iteratively adjusted during the training process (source: MATLAB).

In Fig. 8, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are needed to train a network. However, care should be taken regarding the problem of over fitting during the training process because the ANN may not only learn to approximate the device output but also the measurement/simulation noise in the data. When this happens, the model is known to loose generalization (or its desired predictive) capabilities.

After extracting the bias and weight parameters of the trained networks with two sigmoid functions in the hidden layer, we are able to write the submodels mathematical equation that mimic the nonlinear behavior of the device with a good accuracy. In addition, Fig. 9 and 10 confirm that the feed-forward architecture does have the capability to capture the static nonlinearity of the output buffer without error for both the pull-up and pull-down characteristics.

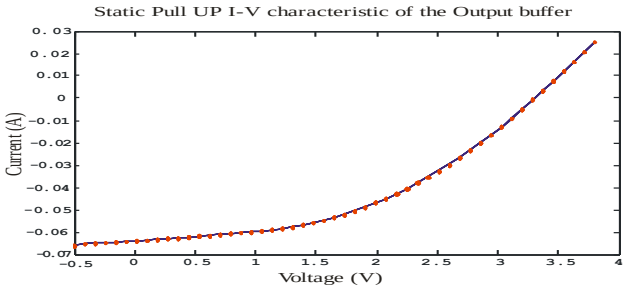

Fig. 9: Function fit for submodel f_1: Transistor-level (straight line) and feed-forward ANN model (dotted line).

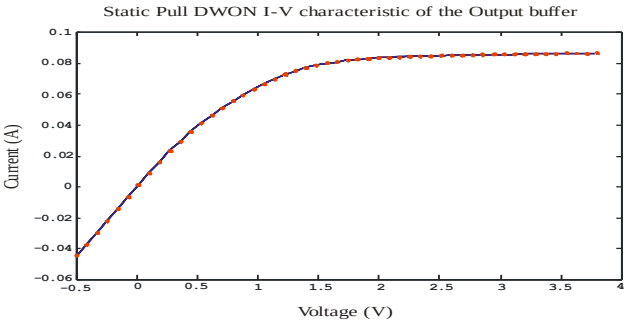

Fig. 10: Function fit for submodel f_2. Transistor-level (straight line) and feed-forward ANN model (dotted line).

However, the driver circuit shows inherent capacitive dynamic effect when it is driven by a signal close to its operating frequency. Therefore, the static model approximation which has no feedback elements and contains no delays, is no longer valid if we want to preserve transistor-level model accuracy. A good practice would be to use a dynamic ANN - for instance the Nonlinear Autoregressive Network with eXogenous inputs (NARX) - that has the ability to model such devices because the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network. The defining equation for the dynamic nonlinear recursive model would then be:

$$i_0(k) = f_n[i_0(k-1), \ldots i_0(k-r_i), v_0(k) \ldots v_0(k-r_v)] \quad (18)$$

The number of previous input and output time instances r_v and r_i required to model the system accurately is dependent on the complexity of the system being modeled. The input-output behavior can be represented by means of the recurrent ANN as illustrated in Fig. 11.

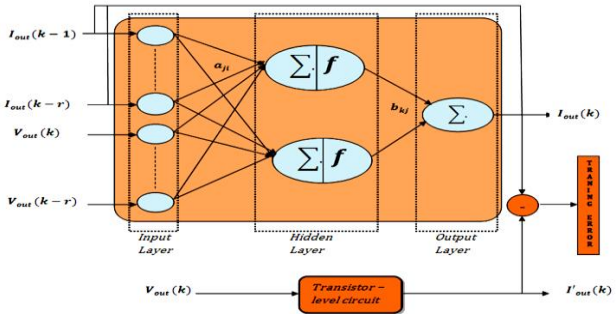

Fig. 11: Scheme used in the training process of the NARX model.

Once submodels f_1 and f_2 are estimated, the switching coefficients, or timing functions, w_1 and w_2 are obtained from the second set of identification signals. Such identification signals are the voltage and current responses recorded during state transitions for two different load conditions. The ideal setup for the generation of such signals is shown in Fig. 12.

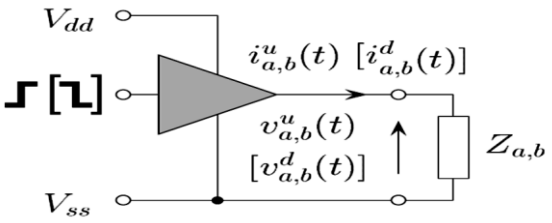

Fig. 12: Simulation setup for the generation of the identification signals for the weight coefficients w_1 and w_2. The current and voltage data are recorded while the driver is loaded by two different loads and is driven to perform an up and down state transitions [4].

For a single Low-to-High (up) transition and for two different port loads (a) and (b), the sequences $(i_u^a(k), v_u^a(k))$, $(i_u^b(k), v_u^b(k))$ are recorded. Their use in (4) leads to the following set of two equations [8]:

$$\begin{cases} i_u^a(k) = w_1^u(k).f_1\left(\Theta_1, x_{1a}(k)\right) + w_2^u(k).f_2\left(\Theta_2, x_{2a}(k)\right) \\ i_u^b(k) = w_1^u(k).f_1\left(\Theta_1, x_{1b}(k)\right) + w_2^u(k).f_2\left(\Theta_2, x_{2b}(k)\right) \end{cases} (17)$$

The elementary weight sequences $w_1^u(k)$ and $w_2^u(k)$ describing the transition can be obtained by simple linear inversion of (17) [8]:

$$\begin{bmatrix} w_1^u(k) \\ w_2^u(k) \end{bmatrix} = \begin{bmatrix} f_1\left(\Theta_1, x_{1a}(k)\right) & f_2\left(\Theta_2, x_{2a}(k)\right) \\ f_1\left(\Theta_1, x_{1b}(k)\right) & f_2\left(\Theta_2, x_{2b}(k)\right) \end{bmatrix}^{-1} \begin{bmatrix} i_u^a(k) \\ i_u^b(k) \end{bmatrix} \quad (18)$$

The same procedure, repeated for a High-to-Low (down) transition, allows to compute two additional elementary sequences $w_1^d(k)$ and $w_2^d(k)$. Finally, a proper concatenation of $w_1(k) = \left[w_1^u(k), w_1^d(k)\right]$ and $w_2(k) = \left[w_2^u(k), w_2^d(k)\right]$ produces the final form of the weight coefficients for a given bit pattern.

Fig. 13 shows an example of the weighting functions generated from 0.35µm-3,3V Austria-Micro-Systems driver model for one resistive load, Z_a, of 50 Ohm and another load, Z_b, composed of a 50 Ohm resistor and a + 3.3 V dc bias.

In principle, there are no restrictions on loads Z_a and Z_b, which can be also real sources stimulating the output port. The best loads would be those allowing $\{i_a, v_a\}$ and $\{i_b, v_b\}$ to explore the widest possible region of the excitation regressor space. Within the class of resistive circuits, it can be proven that the best choice is a resistor for load (a) and the series connection of a resistor and a V_{dd} battery for load (b). Finally, the obtained model is implemented as a subcircuit in any time domain simulator.

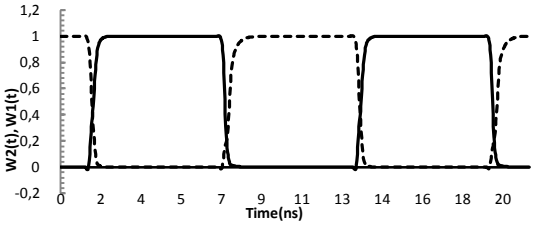

Fig. 13: Weighting functions w_1 (straight line) and w_2 (dotted line) that control the switching between f_1 and f_2.

C. Modeling the Power and ground bounce

Higher levels of integration, higher clock operation frequencies and lower operational voltage make the ground bounce effects more serious in modern chip design processes. In an output buffer circuit, power supply noise and ground noise affect the output voltage and current of the output buffer. Switching of multiple output buffers simultaneously results in large transient currents through the power supply rails which results in SSN. To simplify the SSN analysis, we reduce the block diagram of chip-package to a simpler circuit model alluded to in Fig. 14.

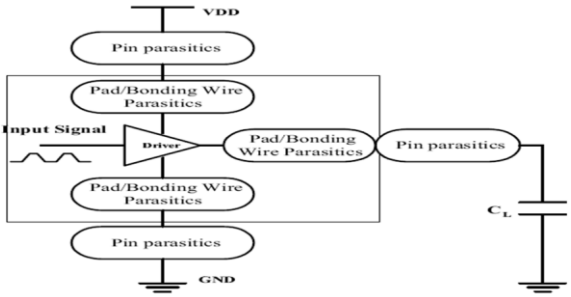

Fig. 14: Block diagram of the chip-package interface.

In fact, the output pad driver is the major contributor to SSN because of large transient currents flowing through the bounding wires, lead frame and pin parasitic inductance. This will cause unwanted fluctuation that can degrade circuit performance or even cause malfunction when the noise peak value exceeds the threshold voltage of the transistor. And this is also the main source for the electromagnetic interference of an IC.

The nonlinear parametric modeling can be extended to multiple ports to capture SSN by adding v_{dd} in the *regressor vector x* (4) as shown in (19):

$$(k) = \begin{Bmatrix} i_0(k-1), \ldots., i_0(k-r) \\ v_0(k), \ldots. \ldots. v_0(k-r) \\ v_{dd}(k), \ldots. v_{dd}(k-r) \end{Bmatrix} \quad (19)$$

The transient responses for the estimation of the model parameters are obtained by driving the model devices with $v_0(k)$ and $v_{dd}(k)$ signals that are multilevel noisy waveforms. The models of the power supply ports are

needed for the simulation of switching noise effects and yield the driver supply current i_{dd} as a function of the supply port voltage v_{dd} and of the output port voltage v, as shown in Fig. 15.

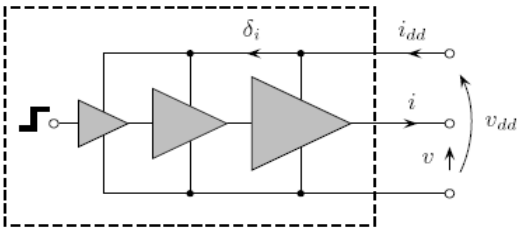

Fig.15: General structure of a driver circuit and its relevant (output and power supply) port electric variables [8].

The power supply port model enabling the SSN simulation is:

$$i_0(k) = w_{d1}(k).f_{d1}[\Theta_1, x(k)] + w_{d2}(k).f_{d2}[\Theta_2, x(k)] + \delta_i(k) \quad (20)$$

where $\delta_i(k)$ takes into account the supply current drawn by the driver stages that precede the last one, and $f_{d1}[.]$ and $f_{d2}[.]$ are the parametric submodels of the current of the last driver stage when it operates in the LOW and HIGH logic states, respectively, and $w_{d1}(.)$ and $w_{d2}(.)$ are the usual weighting coefficients describing state switching.

IV. CONCLUSION

In this paper the previous approaches of IC output buffer macromodeling are revised. Among the available possibilities, we concentrate on the black-box modeling approach via system identification techniques to obtain efficient parametric models whose accuracy must be often compromised with the simplicity of implementation for large scale SI and SSN simulation analysis. An example of a nonlinear parametric model based on SBF expansion is investigated to illustrate the development procedure and to figure out the main advantages and limitations.

The present study is believed to contribute to the systematic discussion of the open research issues of the IC modeling process and so to obtain macromodels that are amenable to overcome the limitations of the methodologies considered so far, such as the IBIS industry standard, the RBF and the SFWFTD methods.

REFERENCES

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

[2] A.K.Varma, M.Steer, P.D. Franzon, "Improving Behavioral IO Buffer Modeling Based on IBIS ", IEEE Transactions on advanced packaging, vol. 31, no. 4, pp.711 - 721 Nov. 2008.

[3] P. Pulici, A. Girardi, G.P. Vanalli, R.Izzi, G. Bernardi, G. Ripamonti, A. Strollo, G.Campardo, "A Modified IBIS Model Aimed at Signal Integrity Analysis of Systems in Package " IEEE Transactions on circuits and systems, vol. 55, no. 7, pp.1921 - 1928 august 2008.

[4] I. S. Stievano, F. G. Canavero, I.A. Maio, "Parametric Macromodels of Digital I/O Ports," IEEE Trans. Adv. Packag, vol. 25, no. 2, pp. 255–264, May 2002.

[5] I.S. Stievano, "Behavioral modeling of nonlinear circuit elements: application to signal integrity and electromagnetic compatibility," Ph.D. Report, Politecnico di Torino, Italy, March, 2001.

[6] B. Mutnury, M. Swaminathan, J. P. Libous, "Macromodeling of nonlinear digital I/O drivers," IEEE Trans. Adv. Packag., vol. 29, no. 1, pp. 102–113, 2006.

[7] B. Mutnury "Macromodeling of nonlinear driver and receiver circuits" Ph.D. Report School of Electrical and Computer Engineering, Georgia Institute of Technology Dec. 2005

[8] I. S. Stievano, I. A. Maio, F.G. Canavero, "M log macromodeling via parametric identification of logic gates," IEEE Trans. Adv. Packag vol. 27, No. 1, pp. 15–23, Feb. 2004.

[9] I. S. Stievano, I. A. Maio, F. G. Canavero, "Behavioral models of i/o ports from measured transient waveforms," IEEE Trans. on Instrumentation and Measurement, vol. 51, no. 6m, pp. 1266–1270, Dec. 2002.

[10] J. C. Pedro, S. A. Maas "Comparative Overview of Microwave and Wireless Power-Amplifier Behavioral Modeling Approaches" IEEE Trans. on Microwave Theory and Tech., vol. 53, no. 4, Apr. 2005.

[11] J. Sjöberg et al., "Nonlinear black-box modeling in system identification: A unified overview," Automatica, vol. 31, no. 12, pp. 1691–1724, 1995.