IBIS and Mpilog Modelling Frameworks for Signal Integrity Simulation

Wael Dghais and Muhammad Alam

Abstract This chapter outlines the structure of the modern integrated circuits (IC) and the necessity of the I/O buffers circuit for ensuring reliable chip-to-chip communication in high-speed digital communication I/O links. Nonlinear circuit simulation based on dc and transient analysis is discussed in order to properly figure out the importance of these algorithms by addressing their convergence issue and computational cost for signal integrity simulation. Moreover, the nonlinear dynamic intrinsic and extrinsic characteristics of the I/O buffers circuit are identified and analyzed. Moreover, this chapter describes IBIS and Mpilog modelling algorithms at circuit level and explains the reason behind the adoption of these modeling tools at different computer aided design (CAD) tools and design houses as opposed to the SPICE transistor level models in the simulation of printed circuit board (PCB) performance for signal and power integrity simulation. The different design steps of the system identification framework are detailed in the context of accurately capturing the nonlinear dynamic behavior of the I/O buffers electrical circuit. With the aim of analyzing the origins of behavioral modeling developed for output buffer/driver to capture its nonlinear and memory effects and in a way to establish a link with their modeling approaches through large-signal equivalent circuit model and parametric curve fitting techniques, a comprehensive overview of the mathematical modeling framework based on system identification theory is presented in this chapter. For sake of simplicity, the methodology is firstly described for one-port active devices. Then, it is extended to two-ports covering the black-box and the gray-box formulations and identification for modeling the driver's nonlinear dynamic behaviors where the state-of the-art Input/output Buffer Information Specification (IBIS) and parametric modeling are analyzed and discussed.

W. Dghais (✉)
Institut National des Sciences Appliquées et de Technologies de Sousse, Sousse, Tunisia
e-mail: waeldghais@ua.pt

M. Alam
School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China
e-mail: alam@ua.pt

M. Alam et al. (eds.), *Real-Time Modelling and Processing for Communication Systems*, Lecture Notes in Networks and Systems 29,
https://doi.org/10.1007/978-3-319-72215-3_2

Keywords IBIS specifications · Mpilog parametric modelling · System identification framework · Signal integrity simulation · Nonlinear dynamic circuit analysis

1 SI Simulation of Digital Transmission Link

Mobile phones, embedded systems, memory interface (DDR4), etc. are integrating an increasing number of sophisticated and heterogeneous functions, such as GPS, video cameras, sensors, etc. This revolution increase of system complexity, the development of new dense 3D packaging structures and the increase of signaling rates [1–5]. The complexity, large integration, and the fast switching signals require the accurate prediction of the whole system performance in the early design stage. The digital chip's processing is carried out within the logic core [1]. However, the I/O buffers are much larger than the core cells and they are required to bond the die to the package and the outside interconnects [5]. This transmission link within the IC introduces distortion into the propagating electrical signals. In fact, it can be decomposed as an active part representing the I/O buffers that interface the logic core and the passive package and drive the multi-conductor interconnected structure of the PCB interconnects as depicted in Fig. 1.

The I/O signals get out from the chip #1 by means of output buffer/driver characterized by a highly nonlinear dynamic behavior. After propagating through IC's package then the PCB, which are usually modeled as passive distributed

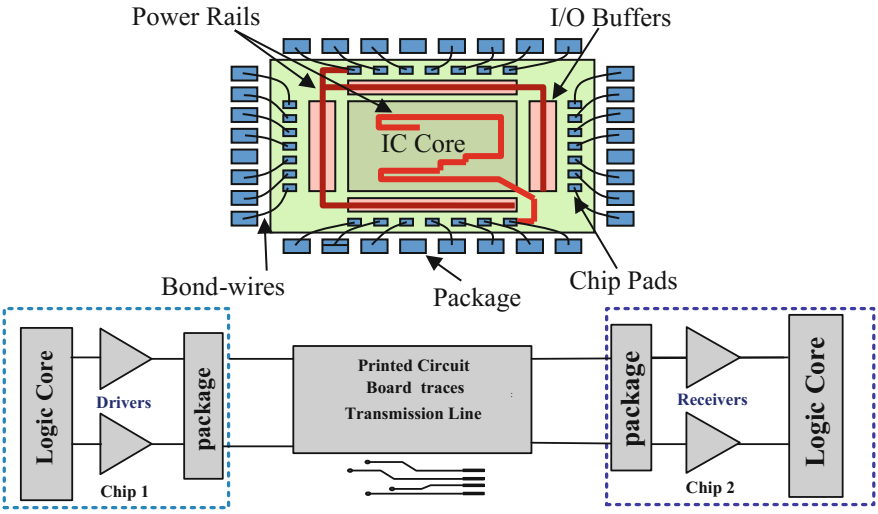

Fig. 1 SI simulation at the PCB level begins with electrical descriptions of the IC die involved, the IC packages, and the traces on the PCB

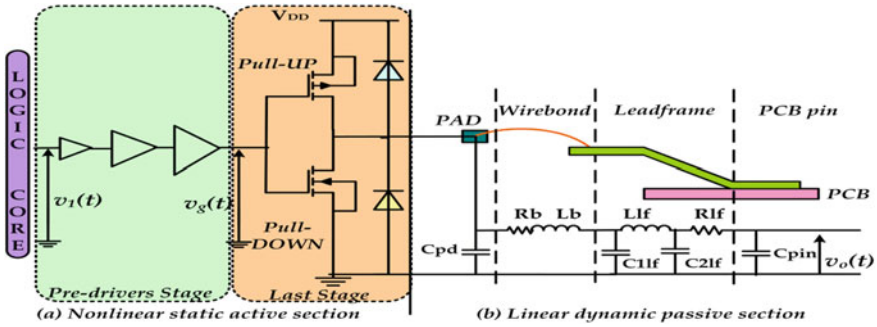

Fig. 2 Pin to die connection in a simple IC. The I/O buffer model sections: **a** Nonlinear active section. **b** Linear package network section

transmission line elements, the signal reaches the chip #2 where the input buffer/receiver detects the transmitted input signal with the above mentioned imperfections according to the switching point voltage. A more detailed description of the I/O buffer enclosed in its package is shown in Fig. 2 [1–5]. The IC die connection pads are connected to the IC pins using very fine wires. The values of lead resistance, lead inductance and package capacitance are different for each pin on the IC because the pin to die connecting wires are different lengths and different geometries [1–4].

The components on the die also have parasitic capacitance associated with them, so the model of say an input pin on even a simple IC, is quite complex. In the active section, the last stage's large transistors are arranged as a pull-up (PU) and a pull-down (PD) network. They impose the dominant electrical behavior of the device, while the pre-drivers stage act mostly as highly nonlinear voltage transfer characteristics followed by a resistive path feeding the gate capacitances of the last stage. On the other hand, the passive section structure is extracted from the associated small-outline IC package category [2–4, 6], from Spectre simulator. The first element, C_{pd}, stands for the capacitance associated to the pad where the bond wire attaches the die to the package. After that, the quantities L_b and R_b are respectively the lumped bond wire inductance and resistance. Besides, the inductance L_{lf}, the coupling capacitances C_{1lf} and C_{2lf} and the resistance R_{lf} represent the lead frame equivalent lumped model. Finally, C_{pin} models the capacitive coupling between the lead frame solder pad and the PCB backplane ground.

Due to the high switching frequency that is always being pushed to higher limits, and to the current intensity that is needed to source a generalized set of external ICs, output buffers are forced, in most cases, to distort the I/O signals due to their nonlinearity and dynamics. Thus, they constitute the IC bottleneck in terms of switching speed. In this concern, SI is a very important task of ensuring sufficient fidelity of a signal transmitted between a driver and a receiver for proper functioning of the circuit (e.g., the signals over the high-speed bus between a processor and its chipset). A signal with good integrity characteristics is defined as one observed at the receiver within the desired time window and with adequate

amplitude levels. The ideal voltage waveform in the perfect logic world and the distorted signal—in terms of amplitude and time deviations—by the active driver and the package dynamics of Fig. 2 are illustrated in Fig. 3 where the different manifestations of SI degradation are [2, 3]:

- A high logic level lower than the minimum high acceptable level, V_{IH},
- A low logic level higher than the maximum acceptable low level, V_{IL},
- Excessive signal delay—a rising/falling (t_r/t_f) edge shows skews longer than the acceptable timing windows,
- Overshoot, undershoot and glitch—ringing, long settling time.

These effects are aggravated with the increase of the driver's clock frequency and the mismatch load with external interconnects. At the receiver's input, voltage above the reference value V_{IH} is considered as logic high, while voltage below the reference value V_{IL} is considered as logic low. Any delay or amplitude's distortion of the waveform will result in a failure of the data transmission. For instance, if the signal in Fig. 3 exhibits excessive ringing into the gray zone while the detection occurs, then the logic level cannot be correctly received.

Signal integrity (SI) analysis aims to ensure reliable high-speed data transmission by evaluating the three main issues. Firstly, the reflections that occur because of interconnect discontinuities such as impedance mismatch, vias, and other line discontinuities. Secondly, the noise induced by neighboring connections (cross-talk), produced in a signal line by other lines as inductive and capacitive couplings. Thirdly, the disturbance on power distribution by switching of the digital devices introduces a noise in a signal line that manifests itself by the voltage drop along the inductive path of the power supply network for the IC and its packaging. This noise is also called power and ground bounce, ΔI (i.e.$V_n = L(di/dt)$)-noise or Simultaneous Switching Noise (SSN) [3–5]. Moreover, SSN is the main sources for the electromagnetic emission of an IC. This is due to the parasitic inductance of the V_{DD} and V_{ss} line cause voltage bounce. To simplify the SSN analysis of the chip-package interface shown in Fig. 4, we reduce the block diagram of chip-package to a simple circuit model.

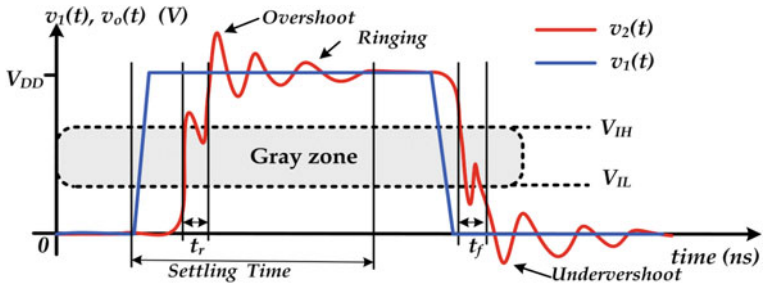

Fig. 3 Ideal the logic core signal and real waveform at receiving gate

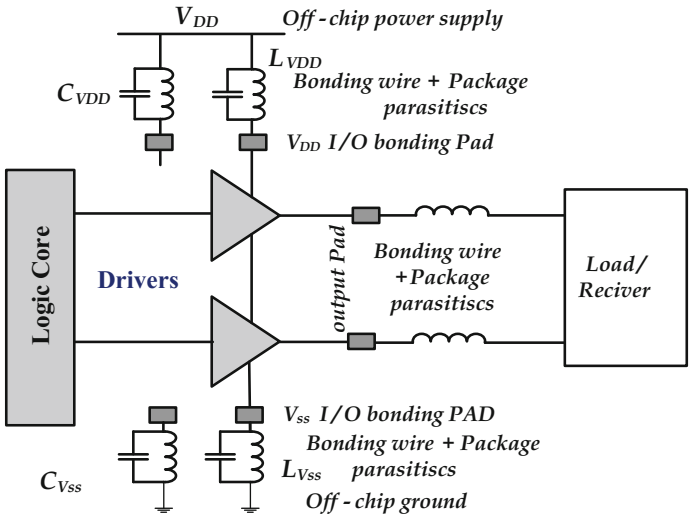

Fig. 4 Die-package interfaces and the electrical equivalent circuit

2 Transient Simulation of Nonlinear Circuits

All SI time-domain tools use iterative circuit-solution techniques pioneered by the simulation program with integrated circuit emphasis (SPICE). An understanding of how SPICE operates therefore tells you a lot about how signal-integrity time-domain analysis algorithms behave. The core of SPICE is the Newton-Raphson algorithm (NRA) which is a guess-and-iterate routine. The dc analysis and algorithm convergence is crucial to perform additional transient, ac or harmonic balance analysis. The study of the dc bias point by SPICE is fundamental to the operation of the other types of simulation. The NRA is a numerical method for solving nonlinear set of equations that describes the electrical circuit behavior based on nodal analysis. NRA consists of linearizing the system to numerically find the root of the function $f(x_{k+1})$. The Taylor series expansion of $f(x_{k+1})$ is [7, 8].

$$f(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0 \tag{1}$$

where $f'(x_k) = \partial f/\partial x$. The iteration step is retrieved so that

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x)} \tag{2}$$

where x_{k+1} and x_k are the values at the next and current iterations, respectively. NRA is recursive $f(x_k)$ and $f'(x)$ needs to be evaluated at each iteration and compared to the previous result in order to determine the direction and the amplitude of the

Table 1 Newton-Raphson algorithm

x_0 = intialguess, $k = 0$
Repeat $x_{k+1} = x_k - \frac{f(x_k)}{f'(x)}$
$k = k + 1$
Until $\|x_{k+1} - x_k\| < threshold$

next iteration until the algorithm converges. Therefore $f(\cdot)$ must be continuous with continuous derivatives. Table 1 describes the main instructions used in NRA.

The extension to the multidimensional case NRA requires the linearization the function relationship $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ relating the input the vector $v_k \in \mathbb{R}^n$ to the output the vector $f(v_k) \in \mathbb{R}^m$ of current or voltage defined branches using the Jacobean matrices. If $v_k \in \mathbb{R}^n$ and F is differentiable at v_k, then its first derivative is given by the Jacobian $J(v_k)$.

$$J(v_k) = \left[\frac{\partial f}{\partial v_1}, \frac{\partial f}{\partial v_2}, \cdots, \frac{\partial f}{\partial v_n}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \cdots & \frac{\partial f_1}{\partial v_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial v_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{3}$$

In this case, the linear map described by $J(v_k)$ is the best linear approximation of $f(\cdot)$ near the point v_k

$$v_{k+1} = v_k - \mathrm{J}^{-1}(v_k)F(v_k) \tag{4}$$

where v_{k+1} and v_k are the voltages at the next and current iterations, respectively. The function describing the circuit behavior has to be monotonic and differentiable and some shape should be avoided in order to ensure convergence even with inadequate guess of the initial conditions as shown in Fig. 5.

For instance, in Fig. 5a, the function is not monotonic, and the selecting of the initial condition has led the NRA to the convergence to the wrong solution. For this reason, it should be better to always question the results of the simulators because they can converge and giving non-physical results. Accordingly, the model equations of the nonlinear electrical elements such as nonlinear current or voltage sources, for diodes, transistors, and memristors must be continuous and differentiable with easy and fast derivative calculation based on the built-in derivative in

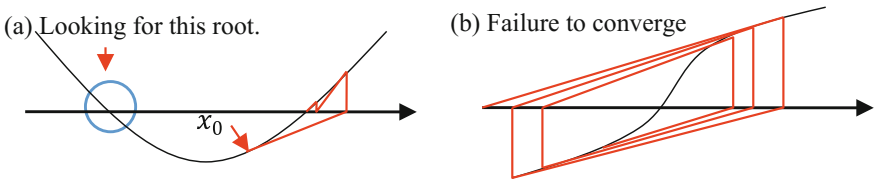

Fig. 5 **a** NRA convergence of non-monotonic function. **b** NRA divergence with monotonic function

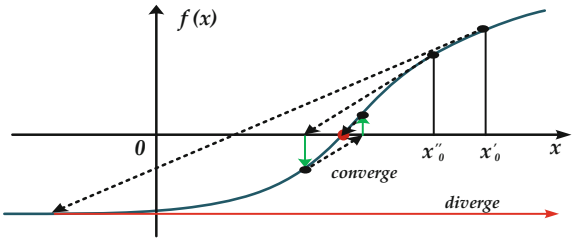

Fig. 6 Dependence of the NRA convergence speed on a good initial guess in a continuous, differentiable and monotonic function

SPICE in order to avoid singular Jacobean calculation. The computation of the dc steady state assumes that voltage and current sources are fixed at their initial values, all capacitances and inductances are considered as open circuits, and short circuits, respectively [7–10]. Moreover, the good initial guess constraint not only the convergence but also the number of iterations to reach the exact the numerical solution as shown in Fig. 6. For instance, the starting guess $x_0^{'}$ was close enough to the solution pointed in red. However, the NRA diverges at the first iteration although the function was smooth and monotonically increasing giving a SPICE error message. An updating of the initial values of the circuit's operating point to x_0" has led to the convergence of the NRA after only two iterations.

Consequently, the improvement of the function modeling the nonlinear two-port and multiport active and passive electronic device is of paramount importance in assessing a high speed convergence of the NRA for dc analysis. Moreover, updating the initial guess can also solve the problem for escalating from the divergence problem as shown in Fig. 7. Once the dc steady sate point is determined for all the node in the circuit, the SPICE algorithm can perform time step integration for transient simulations. In fact, the natural home for nonlinear dynamical phenomena is the time domain, because we can capture transients, and therefore the energy storage or memory effects, as well as the steady state behavior.

The time step integration will be illustrated based on the nonlinear dynamic circuit as shown in Fig. 8. It is composed of a time varying input current source. G is the internal current source conductance, Q_{NL} is a nonlinear capacitance represented as a charge-voltage (Q–V) dependence, and I_{NL} is a nonlinear current source [11].

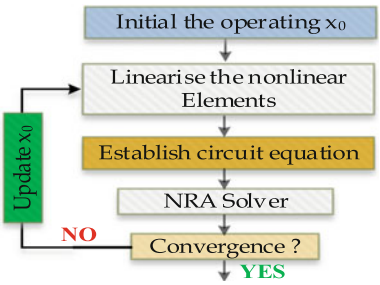

Fig. 7 Flow diagram of the SPICE DC analysis

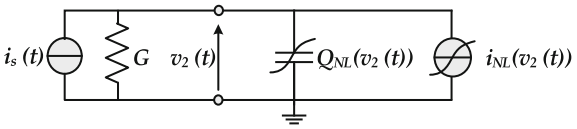

Fig. 8 Circuit schematic considered for transient analysis

The nodal analysis of this circuit leads to the ordinary differential equation, (ODE):

$$Gv_2(t) + \frac{dQ_{NL}(v_2(t))}{dt} + I_{NL}(v_2(t)) = I_S(t) \tag{5}$$

The time-step integration is the most adequate method to numerically solve the ODE of circuit analysis. It only solves directly a discretized version of the ODE (2.5). If the simple forward Euler discretization rule of the time derivatives computed at time instant, t_n, and for a sufficiently small dynamic step $h_n = t_{n+1} - t_n = t_n - t_{n-1}$. This would lead to the following form to explicitly retrieve $v_2(t_{n+1})$ at each time-update:

$$Q_{NL}(v_2(t_{n+1})) = h_n(I_S(t_n) - I_{NL}(v_2(t_n)) - Gv_2(t_n)) + Q_{NL}(v_2(t_n)) \tag{6}$$

That can be written as follows:

$$v_2(t_{n+1}) = Q_{NL}^{-1}[h_n(I_S(t_n) - I_{NL}(v_2(t_n)) - Gv_2(t_n)) + Q_{NL}(v_2(t_n))] \tag{7}$$

Unfortunately, although much simpler and faster to implement and evaluate than the backward Euler rule, this explicit form is rarely used since it usually leads to *stiff* problems. So, the Backward Euler rule is generally preferred:

$$Gv_2(t_n) + \frac{Q_{NL}(v_2(t_n)) - Q_{NL}(v_2(t_{n-1}))}{h_n} + I_{NL}(v_2(t_n)) = I_S(t_n) \tag{8}$$

Which leads to the following implicit form that must be solved with a nonlinear numerical solver as the NRA:

$$h_n(Gv_2(t_n) + I_{NL}(v_2(t_n))) + Q_{NL}(v_2(t_n)) = h_n I_S(t_n) + Q_{NL}(v_2(t_{n-1})) \tag{9}$$

This shows that the circuit's response at any instant t_n is algebraically obtained from the solution of a nonlinear equation, if the excitation is known at that instant, $h_n I_S(t_n)$, as well as the system state, $Q_{NL}(v_2(t_{n-1}))$, or memory as shown in Fig. 9. [5, 11].

The SPICE engine iteratively computes step-by-step the state of the circuit at all future points in time. As the NRA has converged to a satisfactory set of node voltages, $v_2(t_n)$, it evolves to a new time step t_{n+1}, by incrementing the amplitude of the time-varying input signals. The value of the new capacitor voltages and inductor currents are attributed as fixed quantities to the implicit algebraic Eq. (9) and iterates the other node voltages and currents to find a complete solution for circuit

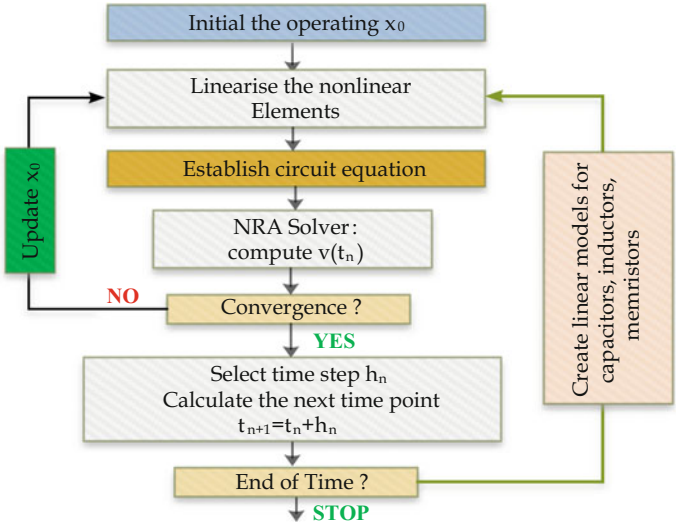

Fig. 9 Simplified SPICE flow of the transient analysis

behavior at time t_{n+1}. The current–voltage relationship across a capacitor, C, $I_c(t) = C\ dV_c(t)/dt$ is discretized, then, the stored voltage $V_c(t_n)$ on a capacitor at time t_n is advanced according to the flowing relationship [1, 5]:

$$V_c(t_{n+1}) = V_c(t_n) + \frac{h_n}{C} I_c(t_n) \tag{10}$$

Similarly, the current flowing, $I_L(t_n)$, through an inductor at time t_n is advanced according to the flowing relationship:

$$I_L(t_{n+1}) = I_L(t_n) + \frac{h_n}{L} V_L(t_n) \tag{11}$$

SPICE updates the new capacitor voltages and new inductor currents as fixed quantities and iterates the other node voltages and currents to find a complete solution for circuit behavior at time t_{n+1}, [7, 11].

2.1 Hierarchy of Modeling Approaches

The SI performance evaluation of high-speed digital systems is usually carried out in a time domain simulation that allows the detailed analysis of the interactions between the digital IC peripheral devices. Hence, Behavioral modeling I/O device circuits to accurately capture their nonlinear dynamic behavior will result in an efficient use of CAD tools by providing the valid and appropriate large-signal

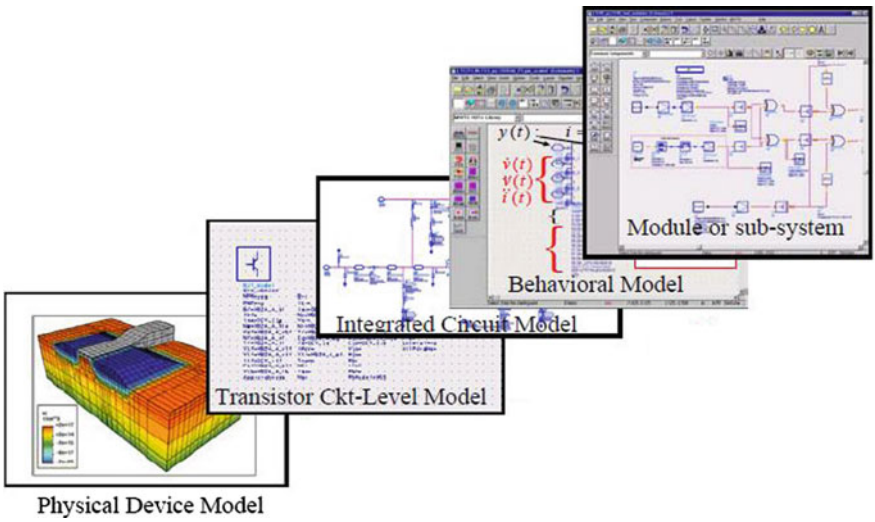

Fig. 10 Model hierarchy according to the physical details included in the model description and the design level copied from [8]

behavioral models of the main connected sub-circuits [7]. Different abstraction modeling hierarchy of the active semiconductor devices is depicted in Fig. 10. The modeling hierarchy begins at the bottom with the TL detailed semiconductor physics model. It ends at the top by a high-level abstraction that describes the terminal behavior through black-box or equivalent circuit behavioral functions. Since there are too many nonlinear interactions taking place at the system-level, the CAD simulation using TL models which is based on NRA takes a long time to converge and uses too much memory to run [8]. A solution to this simulation bottleneck is to generate reduced-order behavioral models that describe the essential behavior of the complex active circuit [9, 10]. The CAD models can be classified according to the level of physical information used in model description.

Firstly, the physical model is the most accurate solution, and usually it is considered as the reference for the device. Unfortunately, the model discloses the internal details of the device. In addition detailed physical descriptions are generally large in size and slows down the simulation. Secondly, the equivalent circuit models are composed of electrical elements such as resistors, capacitors, inductors, and nonlinear current or voltage sources, which can characterize the electrical properties of transistors. Compared with physical device models, the equivalent circuit models require much less computation power. Thirdly, the behavioral model consists in set of mathematical expressions that capture the observed I/O data relationship, describing the essential device behavior based on system identification techniques. The main goal is to enable circuit designers to capture high-level descriptions of components that allow faster simulation and to hide intellectual property. Therefore, circuit designers can send a black-box model to the system

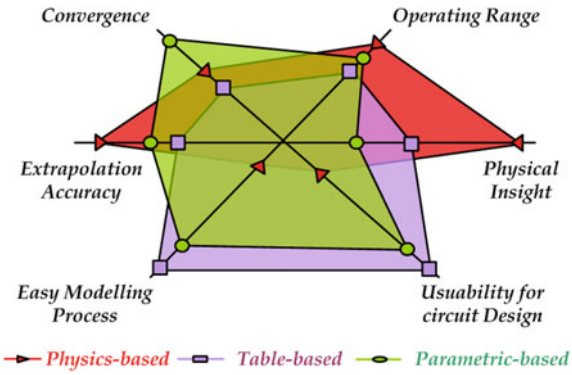

Fig. 11 Types of large-signal CAD tools models

design team reducing overall time-to-market and thus generating additional revenues. A more complete classification can derived, as shown in Fig. 11.

3 System Identification Framework

Behavioral modeling approaches can also be classified according to the level of physical information used in the model mathematical formulation (i.e. physics-based (transparent) or black-box (opaque) model's structure). Both the black-box and gray-box approaches use system identification theory tools to generate their model functions or to extract their parameters [9].

3.1 Device Characterization

The model maker generally has two sources of information; a priori knowledge concerning the I/O buffers (i.e. physics, design circuit, and application) and a representative experimental data of the active device nonlinear dynamic effects. Furthermore, the performance of behavioral model's construction is influenced by two key aspects: the observation and the formulation. The observation refers to the accurate acquisition of the signals at the input and output ports of the device under modeling (DUM) as shown in Fig. 12. The formulation corresponds to the choice

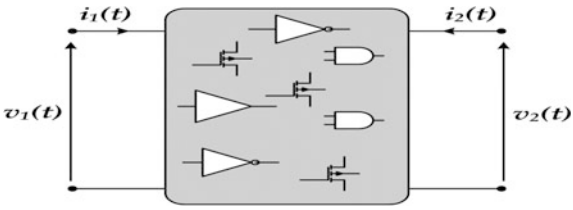

Fig. 12 Generic structure of the two-port model

of a suitable mathematical relation that describes all the significant interactions between the DUM's I/O signals. Accordingly, sufficient background and a priori knowledge of the DUM is required to observe the right behavior and choose the adequate formulation.

The terminal currents $i_1(t)$ and $i_2(t)$ can be expressed in time domain as a function of the two-port terminal voltages $v_1(t)$, $v_2(t)$, and their successive higher order derivatives

$$\begin{cases} i_1(t) = F_1\left(v_1(t),\ \frac{dv_1(t)}{dt},\ v_2(t),\ \frac{dv_2(t)}{dt},\ i_2(t),\ \frac{di_2(t)}{dt},\ \frac{di_1(t)}{dt}\right) \\ i_2(t) = F_2\left(v_1(t),\ \frac{dv_1(t)}{dt},\ v_2(t),\ \frac{dv_2(t)}{dt},\ i_2(t),\ \frac{di_2(t)}{dt},\ \frac{di_1(t)}{dt}\right) \end{cases} \tag{12}$$

The system identification tool does not provide foolproof that always and directly leads to the optimum results. On the basis of the a priori knowledge, the excitation signal is designed to reflect the observed behavior under a specific excitation, the I/O buffer simulation setup is selected, the measurement procedure is specified and the operation range is defined. For instance, the model structure defined by the nonlinear multivariate functions $F_1(\cdot)$, $F_2(\cdot)$ has to be optimized and identified according to the following modeling framework as depicted in Fig. 13 which has an interactive logical flow [12–16]. As arrows point out, the model construction's steps can be recursive and iterative. For instance, if the model's accuracy is not satisfying, various model structures and identification signals should be tested and the process repeated. It is worth noting that the DUM's behavior can be attributed to many sources so that different simulation setups and excitation signal can be used to extract the adequate behavior and then combine them to construct the complete interaction between these effects.

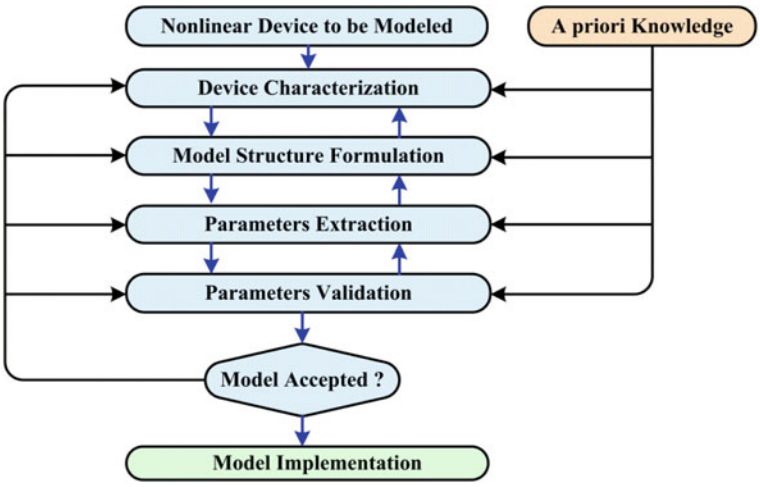

Fig. 13 Behavioral modeling framework based on system identification

The gathered set of measurement results is a good representation of the device behavior only if the identification signal is properly designed cover the frequency and magnitude ranges of interest which is important both for convergence of the behavioral model when used in the simulator. Besides, although the circuit terminations have no impact on the modeling process of linear circuit, the choice of terminations in nonlinear behavioral modeling directly affect the generalization property of the model. In fact, the model accuracy needs to be assessed by subjecting it to various boundary conditions (i.e. I/O signals or terminations). Finally, the resulted model is implemented in the CAD library [8, 15, 16].

3.2 Model Structure Formulation and Extraction

The behavioral modeling activities are not only restricted to replicate the I/O data with general curve fitting methods for given infinite order and complexity [16]. In fact, it involves more challenging tasks as model structure formulation and optimization by means of nonlinear multidimensional function approximation to reduce the model identification and running complexities. This achieved by analyzing the observed I/O signal and investigating the device physical structure and operation in order to develop a model that is coupled with measured quantities. Furthermore, the model extraction heavily depends on the approximation methods used in the model formulation and the followed identification algorithm of its functions and parameters. It is worth noting that not only special care should be taken regarding the simulation setup but also the ability to perform the model extraction from laboratory measurements by collecting the required data from the available equipment and instruments. The recursive or direct model formulation depends on how deep the memory effect is exposed by the DUM behavior excited with suitable excitation signal.

3.3 Model Implementation and Validation

The extracted model must be exported to the format suitable for the use in most modern circuit simulators. An improved behavioral modeling can be achieved by formulating them in language native to the simulator so it eases its implementation and interpolation by different circuit simulators. Besides, the model validation involves various procedures to evaluate how the model relates to the measurement results and the intended usage. It amounts of verifying the correctness and the validity range of the behavioral model in the practical condition application. This requires the selection of metrics to quantify the model accuracy while using boundary conditions that have never been used in its extraction. If the model fails to meet the requirements, the model's identification procedure has to be restarted from an earlier step as shown in Fig. 13. Various factors can lead to a deficient model:

- The driver's measurements do not provide sufficient information or are too noisy.
- The selected model structure (i.e. linear vs. nonlinear) is inappropriate to represent the driver's behavior.
- The chosen order (i.e. number of basis function and the memory length) of the model is too low or too high.
- The model's identification algorithm failed to extract all the parameters correctly.

The model's extrapolation assesses the predictive capabilities of the constructed model by subjecting it to excitation signal's magnitude and frequency ranges higher than the one used in the characterization and extraction steps while loaded by different terminations. In such case, the implemented model's convergence during the transient simulation should be also verified. This is because simulators solve nonlinear algebraic equations using iterative NRA. In fact, the simulator can extrapolate the value of the dependent variable (output of the model) according to the definition given for the functions $F_1(\cdot)$, $F_2(\cdot)$ and cannot encounter the solution of the nonlinear numerical problem and thus will never converge.

4 Driver's Behavioral Modeling Methodologies

Measured or simulated nonlinear behavior can be expressed by a series of equations in time or frequency domain using voltages, currents or incident and reflected waves at all ports. If one performed an infinite number of measurements by changing the environment (i.e. power supply levels, biases, load impedances, etc.), the resulting infinite table of realizations would describe this device completely. However, performing a finite set of observations that can be interpolated with confidence using mathematical modeling and function approximation is more practical.

4.1 Mathematical Modeling and Function Approximation

Parametric behavioral models are based on approximating the device's behavior by a simpler function over some defined region and to some specified accuracy. When a function is fitted to measured data, various candidate basis functions can be selected from a large set, such as the polynomial family, rational functions, splines, and neural networks which are discussed in [11, 16]. In one-port networks, the input and output signals (i.e. voltage, current or incident and reflected waves) that enter and exit the one terminal network as shown in Fig. 14.

In a memoryless or static system,—in which no charge or magnetic flux storage elements (no capacitors or inductors) exist, so that the voltages and currents at any instant do not depend upon previous values of voltage or current-, the output of the

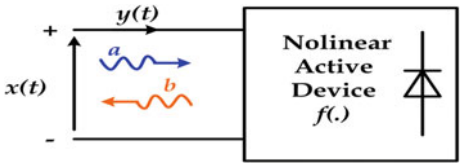

Fig. 14 One-port active network

one-port device, $y(t)$, can be uniquely defined as a function of the instantaneous input signal $x(t)$, and the model can be reduced to $y(t) = f(x(t))$ or $y = f(x)$ since the dependence on time is immaterial. However, the output buffers, as all active electronic circuits, present memory effects. The output now depends also on the input, or output past and system state. The I/O relation becomes an operator that maps a function of time $x(t)$ onto another function of time $y(t)$. Thus, the input–output mapping of the device can be represented by the following forced nonlinear ordinary differential equation [17]:

$$f\left(y(t), \frac{dy(t)}{dt}, \ldots, \frac{d^p y(t)}{dt^p}, x(t), \frac{dx(t)}{dt}, \ldots, \frac{d^r y(t)}{dt^r}\right) = 0 \tag{13}$$

This states that the output and its time derivatives can be nonlinearly related to the input and its time derivatives. By assuming the back-ward discrete-time approximation of the continuous-time derivative of a signal (t):

$$\frac{dz(t)}{dt} \cong \frac{1}{T_s}(z(kT_s) - z(kT_s - T_s)) = \frac{1}{T_s}(z(k) - z(k-1)) \tag{14}$$

the discrete time version of (2.1) can be expressed in the recursive form:

$$y(k) = f_R[y(k-1),\ y(k-q_1),\ x(k),\ x(k-1),\ x(k-q_2)] \tag{15}$$

The sampling period is T_s. The present and sampled output y(k) depends in a nonlinear way, on the output past $y(k - q_1)$, the present input $x(k)$, and its past $x(k - q_2)$. Nevertheless, such a system; fading memory, can also be represented with any desirable small error by a non-recursive, or direct form, where the relevant input past is restricted to $q\{0,\ 1,\ 2,\ \ldots,\ q_D\}$:

$$y(k) = f_D[x(k),\ x(k-1),\ \ldots,\ x(k-q_D)] \tag{16}$$

Similarly, the behaviors of active devices, (e.g. I/O buffers) encountered in the high-speed digital I/O link are described by nonlinear ODE. The behavioral modeling task amounts of capturing the essential observed dynamics by constructing the function,$f(\cdot)$, which maps the $(q_2 + q_1 + 1)$ independent variables into the dependent variable. This function can have a direct or recursive formulation, $f_D(\cdot)$ and $f_R(\cdot)$, respectively. The following subsections presents the parametric and the equivalent circuit model used in representing the nonlinear functions $f_D(\cdot)$ and $f_R(\cdot)$, respectively [16, 17].

4.2 Parametric Functions

In the first case, the direct form of a memory polynomial that interpolates $f_D(\cdot)$ functions can be written as:

$$\begin{aligned} y(k) &= \sum_{q=1}^{Q} a_1(q)x(k-q) + \\ &\sum_{q_1=0}^{Q}\sum_{q_2=0}^{Q} a_2(q_1,\ q_2)x(k-q_1)x(k-q_2) + \ldots \\ \sum_{q_1=0}^{Q}\ldots\sum_{q_N=0}^{Q} &a_N(q_1,\ \ldots,\ q_N)x(k-q_1)\ldots x(k-q_N) \end{aligned} \tag{17}$$

where the $a_N(q_1, \ldots, q_N)$ are the polynomial coefficients that can be estimated in a direct way by a simple least-squares method. The last equation can be written in matrix form $Y = F \,.\, A$. In which A is the vector of the model coefficients, Y is the vector of the output data samples and F is the matrix that contains the mixture elements of the input signal x. The least-squares method can be used to find the polynomial coefficients by:

$$A = \left(F^H F\right)^{-1} F^H Y \tag{18}$$

where F^H denotes the conjugate transpose of the matrix F. Although simple in concept, this finite impulse response (FIR) polynomial model architecture is known for its large number of parameters. For similar approximation capabilities and many fewer parameters, the recursive polynomial infinite impulse response (IIR) structure can be used, but its stability should be verified at the model extraction and validation steps. In the second case, the functions $f_D(.)$ and $f_R(.)$ are built using the ANNs. The recursive structure is illustrated in Fig. 15 in which the dynamic mapping depends on the past values of both the I/O signals. The ANN input-output mapping, with N neurons, and $\{q_1, q_2\}$ I/O dynamic order is mathematically described by:

$$\begin{cases} u_n(k) = \sum_{q=1}^{q_1} \alpha_{yn}(q)y(k-q) + \sum_{q=0}^{q_2} \alpha_{xn}(q)x(k-q) + \beta_n \\ y(k) = \beta_o + \sum_{n=1}^{N} \alpha_{yo}(n)\phi(u_n(k)) \end{cases} \tag{19}$$

where $\alpha_{yn}(q)$, $\alpha_{xn}(q)$, $\alpha_{yo}(n)$, are weighting coefficient, β_n and β_o are bias parameters and $\phi(\cdot)$ is the activation function. Figure 15 shows that the model output $y(k)$ is built from the addition of the activation functions $\phi(u_n(k))$ and the weighted outputs. The input $u_n(k)$are biased sums of the various delayed version of the input x(k)

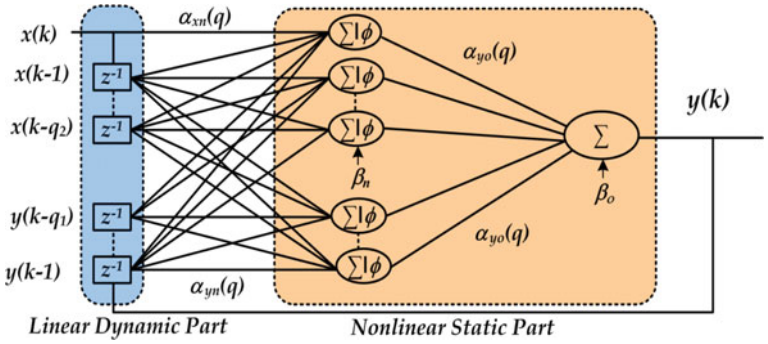

Fig. 15 General structure of a recursive ANN

and output $y(k)$ weighted by the coefficient $\alpha_{xn}(q)$, and $\alpha_{yn}(q)$, respectively. The ANN model is nonlinear in the parameters $\alpha_{xn}(q)$ and $\alpha_{yn}(q)$ and linear in the parameters $\alpha_{yo}(n)$, and β_o. Various nonlinearities, $\phi(\cdot)$, can be used as activation functions. For instance, the hyperbolic tangent function is defined as:

$$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{20}$$

The ANN's activation functions are bounded in output amplitude. Thus they do not share the catastrophic degradation of polynomials outside the training zone, during circuit simulation that could escalate to divergence problems. Since, the ANNs model is usually considered as nonlinear with respect to all its parameters [16, 17], a nonlinear least squares algorithm is used for training by minimizing the mean square error (MSE) cost function between the model and the TL outputs as shown in Fig. 16.This means to find:

$$\Theta|\min\left\{\frac{1}{N}\sum_{k=1}^{N}(\hat{y}(k) - y(k))^2\right\} \tag{21}$$

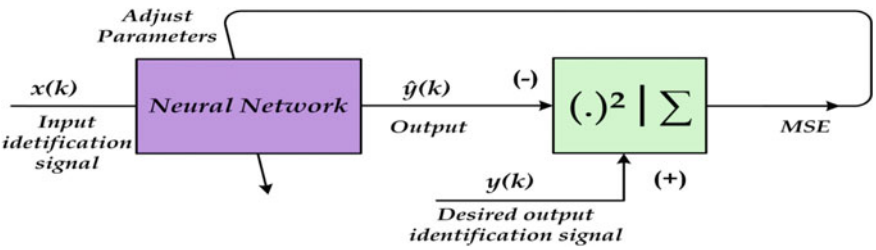

Fig. 16 Training scheme of an artificial neural network

where $\hat{y}(k)$ is the sampled output signal, N is the total number of samples and $y(k)$ is the model response to the sampled input signal $x(k)$. It is worth to note that the problem (21) is nonconvex and the solution obtained by means of the optimization algorithm can get stuck in local minima. Thereby, a good initial estimate of the model's parameters is crucial, since the initial starting point determines in which local minima the algorithm will end up which will affect the model generalization capabilities.

The ANNs and the polynomial expansion have been proven to universally approximate any continuous multivariate function with any degree of accuracy provided that enough neurons and dynamic orders are considered which defines the number of the ANN's weights and bias coefficients, and so its complexity. However, there is no way to know a priori the numbers of neurons or dynamic orders required for representing a specific system or the modeling improvement gained when these parameters are increased. Beyond that, the instability issues that the recursive ANNs may show while the training or circuit simulation, the ANNs model's identification may suffer from other drawbacks involving over-fitting and the difficulty to physically interpret the model itself.

4.3 Scattering Functions

For linear dynamic devices (e.g. filters, package, etc.), S-parameters provide a black-box behavioral model in frequency domain. They are small signal representation of the transmission and reflection coefficients of a DUM at the input and output ports, and are defined in terms of a given reference source and load impedances, usually 50 Ω, as shown in Fig. 17.

The two-port reflected waves b_1 and b_2 can be related to the incident waves a_1 and a_2 at the I/O ports, respectively, by the S-parameter matrix:

$$\begin{cases} b_1 = S_{11}a_1 + S_{12}a_2 \\ c \\ b_2 = S_{21}a_1 + S_{22}a_2 \end{cases} \equiv \begin{bmatrix} b_1 \\ c \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ c & c \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ c \\ a_2 \end{bmatrix} \tag{22}$$

In the large-signal case some of the energy is reflected and transmitted at harmonics of the fundamental frequency. The S-parameters can be extended as a

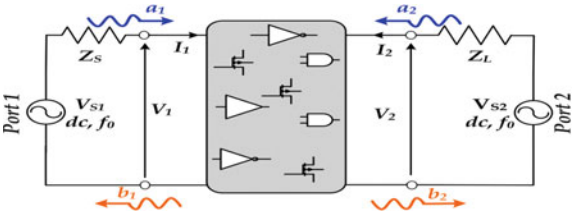

Fig. 17 Two port S-parameters characterization for the active device

function of the dc bias point and the measurement frequency. We can also introduce the port indices i and j and write the two port equations in summation. The S-parameter dependence on bias point and frequency f_0 of the source frequency is expressed as:

$$b_i = \sum_{i=1}^{2} S_{ij}(V_{DC}, f_0)a_j \tag{23}$$

The recorded large-signal and frequency-domain data from the TL simulation, or tailored measurement setup performed with a Vector Network Analyzer (VNA), of the active DUM (e.g. a diode or MOSFET transistors) can be fitted to an equivalent circuit model in terms of resistors, capacitors, inductors, and voltage controlled current sources or voltage controlled voltage sources, etc. In fact, the measured S-parameter data are converted into Y-parameters. Then, the circuit is extracted by fitting the data into the adequate equivalent network structure for the input and output ports. Finally, the overall equivalent circuit can be run in a SPICE based tool, and the S-parameters can be calculated and compared with the measured values. The implementation of this extraction technique is outlined in Fig. 18.

The physical knowledge can help to further reduce the model order and achieve a lower complexity by only maintaining the necessary information needed to describe the DUM. Qualitative and quantitative considerations on the validity of the circuit models extracted can be drawn by simulating the large-signal frequency domain (e.g. Harmonic Balance) to compare the output spectrum and transient response (e.g. SPICE) to compare the time domain signal or the eye-diagrams. The reduced-order and physically inspired model will help the semiconductor designer to evaluate the impact of the device characteristics affecting the communicating signals. These measurement-based models are usually implemented as simple look up tables (LUT) which are interpolated for any input condition to perform SI

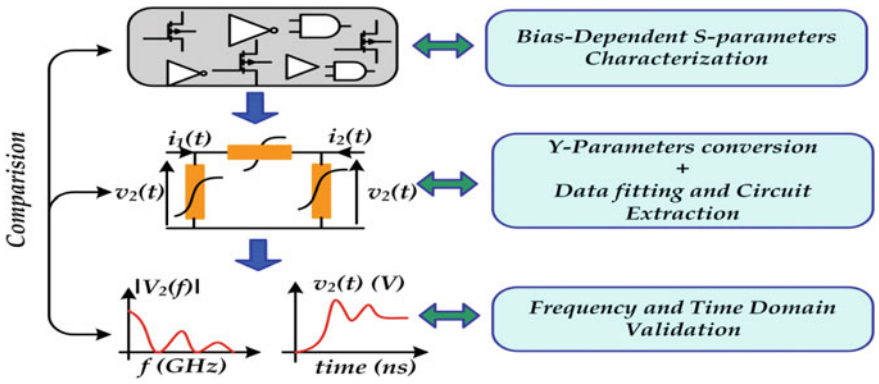

Fig. 18 Implementation diagram of the equivalent circuit extraction technique based on frequency domain measurements

simulation. The constructed model should balance the trade-off between accuracy, parameter estimation, and running complexity.

4.4 Equivalent Circuit Based Functions

An alternative approach to describe model structure is based on the equivalent circuit or the device physical properties. The equivalent circuit behavioral models can be seen as gray box modeling approach which naturally produces more efficient behavioral model by enhancing the accuracy level and reducing the complexity of the black-box model. The driver's behavioral models must be able to represent in the time domain the port's voltage and current or incident and reflected wave relationship in frequency domain. Therefore, their topology is of double-input double-output nature. This is a generalization of single-input single-output recursive or direct models of Eqs. (3) and (4) of the one-port first example. If the input and the output are components of the incident and reflected waves, then Eqs. (3) and (4) represent a nonlinear generalization of the linear scattering matrix. If they are stated as voltages and currents, we end up with a nonlinear generalization of the nonlinear admittance or impedance matrix formulation. To illustrate the principle, the two-port large-signal model of a MOSFET is considered as represented in Fig. 19.

The simplified scheme consists of the parallel connection of a charge and current source. They are the device's state functions that reproduce the two ports electrical behavior and can be extracted based on equivalent circuit or black-box model identification providing large-signal data from time domain or frequency domain measurements [18, 19]. The terminal currents can be expressed in the time domain by:

$$\begin{cases} i_1(t) = i_1^c(v_1(t),\ v_2(t)) + i_1^d(v_1(t),\ v_2(t)) \\ \qquad\qquad c \\ i_2(t) = i_2^c(v_1(t),\ v_2(t)) + i_2^d(v_1(t),\ v_2(t)) \end{cases} \tag{24}$$

and the displacement current at the I/O ports are defined as follows:

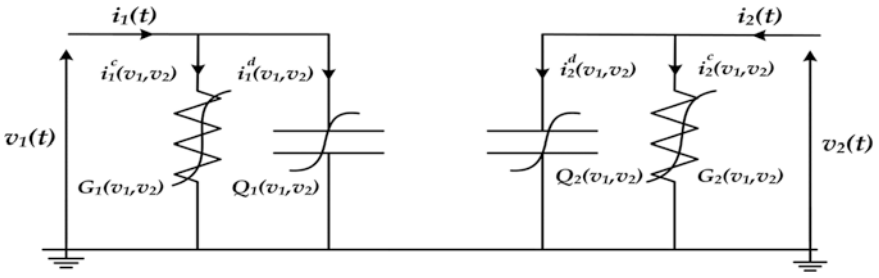

Fig. 19 Nonlinear equivalent circuit model [19]

$$\begin{cases} i_1^d(v_1(t),\ v_2(t)) = \frac{dQ_1(v_1(t),\ v_2(t))}{dt} \\ i_2^d(v_1(t),\ v_2(t)) = \frac{dQ_2(v_1(t),\ v_2(t))}{dt} \end{cases} \tag{25}$$

Defining: $C_{ij}(v_1(t),\ v_2(t)) = \partial Q_i(v_1(t),\ v_2(t)) \ / \ \partial v_j$, (24) is rewritten as:

$$\begin{cases} i_1(t) = i_1^c(v_1, v_2) + C_{11}(v_1, v_2)\frac{dv_1(t)}{dt} + C_{12}(v_1, v_2)\frac{dv_2(t)}{dt} \\ i_2(t) = i_2^c(v_1, v_2) + C_{21}(v_1, v_2)\frac{dv_1(t)}{dt} + C_{22}(v_1, v_2)\frac{dv_2(t)}{dt} \end{cases} \tag{26}$$

The nonlinear functions $i_1^c(\cdot)$, $C_{11}(\cdot)$, $C_{12}(\cdot)$, $C_{21}(\cdot)$ and $C_{22}(\cdot)$ that describe the two port's large-signal model are single-valued. This results into a set of two equations with six unknowns, which can be solved if we have three distinct (independent) measurements by which the instantaneous terminal voltages remain unchanged. This imposes special experimental conditions on the large-signal measurements, such as the presence of a load pull system [19]. The extraction of C_{11} and i_2 have shown a good agreement with reference results, obtained by standard dc and S parameter measurements on the same device [20]. In this equivalent circuit model there is no feedback element that represents the dependency of the input and output port variables. In general, this feedback element is reflected by the recursive model formulation.

5 Digital I/O Buffer Behavioral Modeling Approaches

The most popular driver behavioral models is the large-signal equivalent circuit IBIS model. Nevertheless, with the continued transistor scaling and the increase of the maximum operating frequency, other approaches, which are based on parametric modeling, appeared to complement the IBIS model and proved to be more accurate in SI simulation. This section presents the linear interpolation concept for bi-dimensional function and the generation procedures of the IBIS and parametric modeling [11–13] are presented as an illustration of the gray-box and black-box approaches.

5.1 Mathematical Formulation

The driver's basic structure where the last stage comprises the largest PU and PD transistors, with power and ground clamping diodes for electrostatic discharge (ESD) protection, as shown in Fig. 20. Assuming a constant power supply voltage, V_{DD}, the two port driver have quite distinct characteristics: while the input of the predriver is controlled by the IC core circuitry, whose output is just a data signal of quantized values "1" or "0", the output port drives high output currents i_2 (when

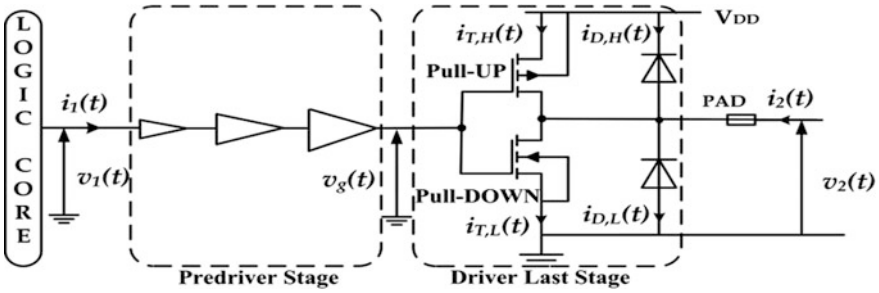

Fig. 20 Basic structure of the two port I/O buffer

compared to the almost negligible, and non-accessible, current i_1 of the input port) due to the large size of the last stage transistors.

For this reason, the model concentrates only on the output port's nonlinear and dynamic voltage-current relationship, adopting the following event-driven and time domain form:

$$i_2(t,\ n) = F(v_1(n),\ v_2(t),\ dv_2(t)/dt,\ \ldots,\ di_2(t)/dt,\ \ldots) \tag{27}$$

where $F(\cdot)$ is a suitable multidimensional nonlinear function. The event n defines the up and down transitions that mark the change between the driver's High, "*H*", and Low, "*L*", logic states. In order to ease the process of modeling and simulation, we will use an interpolation technique [21] for approximating the nonlinear multi-dimensional function $F(\cdot)$, which leads to the following model structure:

$$i_2(t) = w_L(t).i_L(v_2(t),\ d/dt) + w_H(t)\ .\ i_H(v_2(t),\ d/dt) \tag{28}$$

where *d/dt* represents the dependency on the successive derivatives of the output currents, $i_L(t)$ or $i_H(t)$, and of the output voltage $v_2(t)$. The local models $i_L(\cdot)$ and $i_H(\cdot)$ are appropriate multi-input single-output nonlinear functions. They model the nonlinear dynamic admittance of the PU and PD devices of the driver's last stage. The output current of the local models are scaled using the stepwise weighting functions $w_L(t)$ and $w_H(t)$ that capture the timing behavior of the variable $v_g(t)$, as the IC core sends the bit patterns '01' and '10' (i.e. the driver is changing from the *L* to *H* states for the up transition and from the *H* to *L* states for the down transition). Assuming that the interpolation is linear, so that, the switching between the PU and PD devices is ideal and symmetric, the timing functions are complementary [21]

$$\begin{cases} w_L(t) = \frac{v_g(t) - v_{g,L}}{v_{g,H} - v_{g,L}} = \frac{v_g(t)}{V_{DD}} \\ w_H(t) = 1 - w_L(t) \end{cases} \tag{29}$$

The driver last's stage can be considered as a unilateral device such that the effect of v_2 variation, due to the loading PCB traces impedance mismatch, on the v_g is negligible. The last stage buffer's current-voltage relationship can be

approximated by model (30) relating $i_2(t)$ and $v_g(t)$ and their successive derivative based on the observed I/O signals from transient simulation, it is fair to assume that:

$$i_2(t) = H\left(v_g(t), \frac{dv_g(t)}{dt}, \frac{d^2v_g(t)}{dt^2}, \frac{d^3v_g(t)}{dt^3}, \ldots\right) \tag{30}$$

where $H(\cdot)$ is the large-signal current source function that describes the voltage transfer characteristics (VTC) of the driver last stage while it drives a load. According to the polarity of the buffer (inverting/not inverting). Thus, the functions $w_L(\cdot)$, $w_H(\cdot)$ and $i_L(\cdot)$, $i_H(\cdot)$ can be extracted based on the observation of accessible output signals. Conceptually, the two previous driver's behavioral modeling approaches [11–15] share the same model structure (28) but only differ in the mathematical formulation and the identification process for modeling the local models, $i_L(\cdot)$ and $i_H(\cdot)$. Firstly, IBIS separately extracts the nonlinear static function which is implemented as LUT and then the dynamic is assumed to be linear and modeled as lumped capacitor. However, the last stage large-signal behaviors were simultaneously extracted in the parametric approach using dynamic I/O data that was fitted by ANNs or spline functions [11, 12].

5.2 IBIS Behavioral Modeling

The IBIS model is a set of specifications describing the I/O buffers model extraction and formatting data based on simplified equivalent circuits [22]. The IBIS driver's description is generally divided into four elements as shown in Fig. 21 [22]. These are the PU and PD transistors, ESD clamp diode with current–voltage (I–V) dc static table. Then, to better capture the interpolation between local models ($i_L(\cdot)$ and $i_H(\cdot)$), IBIS version 2.1 provides the transient behavior at up and down transitions by the means of timing voltage (V-t) data. Also IBIS provides the

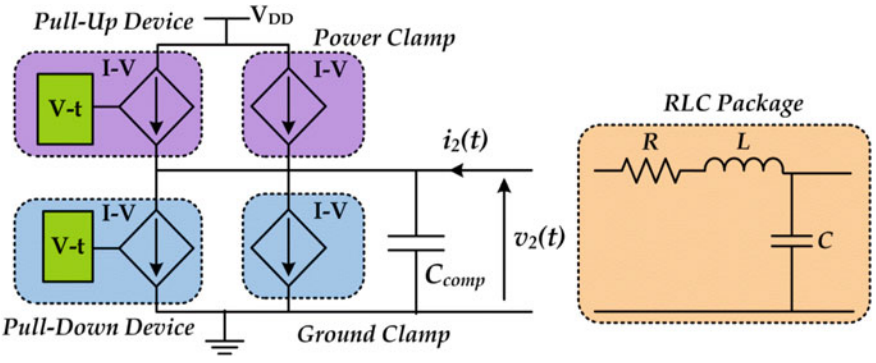

Fig. 21 Key portions of the active intrinsic parts of IBIS [22]

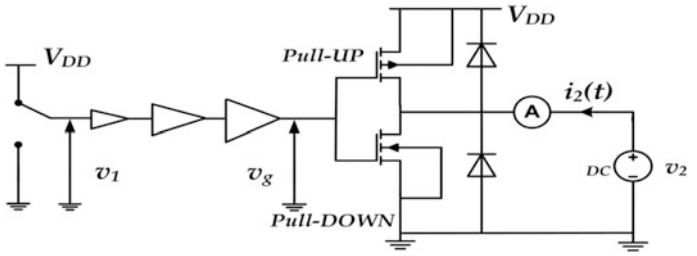

Fig. 22 Simulation setup for the generation the four I–V curves for IBIS

lumped RLC package model and the pad capacitor. These elements are represented as text files.

In order to generate the four I–V curves that describe the IBIS driver's last stage, we slowly increase the voltage and measure the current and voltage with an ammeter and voltmeter connected at the driver's output port as shown in Fig. 22. They are PU, PD, ground clamp and power clamp curves. The PD curve is a result of subtracting the ground clamp I–V curve from the logic-low I–V curve, since this is where the PD transistor is active as shown in Fig. 23a. Similarly, the PU curve is generated by subtracting the power clamp I–V curve from the logic-high I/V curve, since this is where the PU transistor is active as shown in Fig. 23b. The full range of the measurement is from $-V_{DD}$ to $2V_{DD}$ which is the possible range of voltages that the output could see in a transmission line environment.

The ground clamp curve is derived from the ground relative data gathered while the driver is in the high impedance state and illustrates the region where the ground clamp diode is active. The range is from $-\boldsymbol{V_{DD}}$ to $\boldsymbol{V_{DD}}$. The power clamp curve is derived from the $\boldsymbol{V_{DD}}$ relative data gathered while the buffer is in a high impedance state and shows the region where the power clamp diode is active. This measurement ranges from $\boldsymbol{V_{DD}}$ to 2 $\boldsymbol{V_{DD}}$. The PU and power clamp curves are $\boldsymbol{V_{DD}}$ relative, meaning that the voltage values are referenced to the $\boldsymbol{V_{DD}}$ pin since the currents

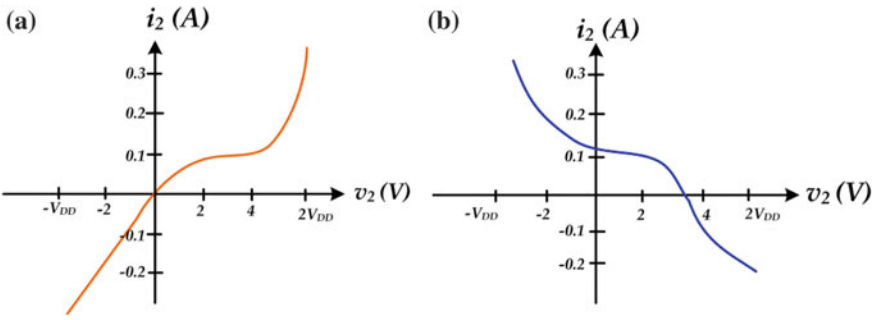

Fig. 23 IBIS output buffer static I–V characteristics **a** PD I–V, **b** PU I–V

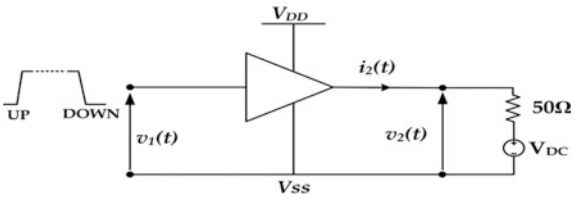

Fig. 24 Simulation setup for the transient identification signals $\{i_2(t), v_2(t)\}$

depends on the voltage between the output and $\boldsymbol{V_{DD}}$ pin and not the voltage between the output and ground pins.

IBIS model also provides rising and falling V-t waveforms. These curves can be recorded from SPICE simulations when the buffer output is terminated with 50 Ω and dc voltage source, V_{DC}, for the up and down step excitation at buffer input as shown in Fig. 24. The dc I–V curves and the timing data are used to calculate the step input describing functions (SDFs), where the effect of package parasitic are ignored. The SDFs are computed using linear inversion.

$$\begin{bmatrix} w_H(t) \\ w_L(t) \end{bmatrix} = \begin{bmatrix} i_{L,V_{ss}}(v_2) & i_{H,V_{ss}}(v_2) \\ i_{L,V_{DD}}(v_2) & i_{H,V_{DD}}(v_2) \end{bmatrix}^{-1} \begin{bmatrix} i_{2,V_{ss}}(t) \\ i_{2,V_{DD}}(t) \end{bmatrix} \tag{31}$$

The model (28) can be adapted to a one waveform condition by considering assumption (29). In order to maximize the correlation of the complete behavioral model, IBIS adds the linear capacitive effect to model the additional output dynamic of the driver's last stage [22–25]. Besides, the predriver's nonlinear dynamics are encapsulated in the timing behavior of the extracted four tables V-t. Data provided by IBIS, in accordance to the assumed equivalent circuit, must be translated into an executable model in order to be used in circuit simulation environments. IBIS driver's model is popular and widely used as it is commercially available, has large sets of libraries, and run faster than actual driver's TL models. However, IBIS models have inherent limitations. Most importantly, the IBIS model structure is derived under some assumptions, thus, it can work only for limited data rate input range. In fact, the hard algorithm implementation of the SDFs should provide the synchronization and concatenation of the elementary timing functions to reproduce the desired bit sequence. The model's formulation, extraction, and implementations issues in IBIS generation algorithm limit its use under high data rate excitations. This issue is called overclocking [24]. Besides, the physical effects to be considered are decided a priori when the simplified equivalent circuit is defined. For instance, IBIS models fail to accurately capture the driver's nonlinear memory effects as it relies on static characteristics and lumped linear capacitance C_{omp} [14].

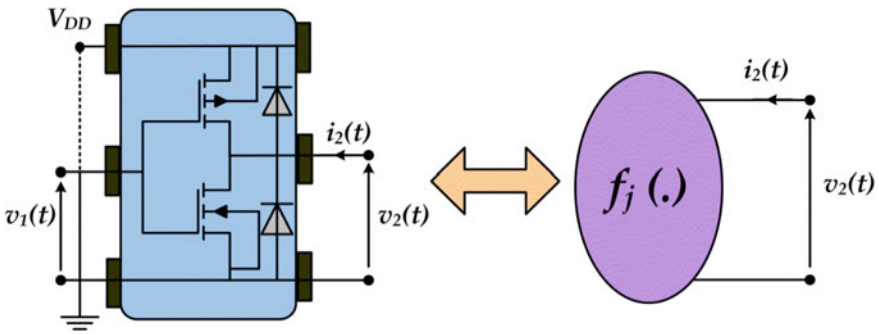

Fig. 25 A functional form reproducing an observed nonlinear dynamic electrical behavior of the device

5.3 Black-Box Parametric Modeling

A second possible alternative to driver's behavioral modeling is via parametric curve fitting techniques as illustrated in Fig. 25. The parametric approaches [11–13] confirm the importance of using nonlinear dynamic local models to develop more effective behavioral model [23–26]. They use ANNs to mimic the nonlinear dynamic I/O data collected by exciting the output port with a piecewise linear voltage with various amplitudes and rise/fall times, along with superimposed white noise [23–26]. Thus, the model adopts a discrete-time recursive structure of the form while the input is kept at H and L logic states:

$$\begin{cases} i_j(t) = f_j\big(\Theta_j,\ x_j(k)\big);\ \ j = L,\ H \\ x_j(k) = \big[i_j(k-1)\ \ldots,\ i_j(k-r),\ v_2(k)\ \ldots,\ v_2(k-r)\big]^{\mathrm{T}} \end{cases} \tag{32}$$

The regressor vector x_j collects the instantaneous I/O signals, and their delayed versions. Θ_j is the vector of unknown parameters, and $f_j(\cdot)$ is a generic nonlinear scalar function modeled by recurrent ANNs with various activation functions such as radial basis function (RBF) [23], sigmoid basis function (SBF), polynomial functions as the spline function with finite time difference [25] or rational basis functions [15]. The advantage of this approach relies in the flexibility of ANNs against the rigid representation adopted by IBIS [22]. Thus the modeled device is considered as a black-box and does not disclose the IP. The nonlinear function $f_j(\cdot)$ interpolates the actual performance of the circuit from the observed behavior which can be obtained from measurements or simulation data. Moreover, the basis functions used in the parametric models should be complete to approximate device's complex nonlinear behavior. Besides, it is preferable that the set of functions composing $f_j(\cdot)$ are orthogonal so the model's parameters are extracted independently. In order to obtain the optimum model which use less neuron and time delays to accurately approximate the function $f_j(\cdot)$, many training data sets have to be tried.

The design of the excitation signals is a matter of repeated estimation experiments. The described parametric approaches [23–25] follow back-box identification by treating the number of state variables as an unknown. However, the physical knowledge via the equivalent circuit topology can be used to prune and reduce the order of the parametric model. As an example, in the case of a one-port nonlinear resistor, $i_1(t) = f_{D1}(v_1(t))$ and in the case of a nonlinear capacitor $i_1(t) = f_{D1}(v_1(t),\ dv_1(t)/dt)$. As the derivative is related to the concept of a time delay and by considering, the terminal currents can be expressed as a function of the terminal voltages and their first derivatives. This can be generalized for an electrical two-port by discrete form equations

$$\begin{cases} i_1(k) = f_{D1}[v_1(k),\ v_1(k-1),\ v_2(k),\ v_2(k-1)] \\ i_2(k) = f_{D2}[v_1(k),\ v_1(k-1),\ v_2(k),\ v_2(k-1)] \end{cases} \tag{33}$$

Then, the modeling problem is simplified by the equivalent circuit knowledge (24) and it is transformed to find the functional relationships $f_{D1}(\cdot)$ and $f_{D2}(\cdot)$ by fitting the measured time domain terminal currents to the already determined independent and state variables. In the general case, the model complexity can be reduced by determining the optimum sets of independent variables and states to predict accurately the device dynamics from the collected time series data which is representative of the device behavior. The initial set of independent variables from which the model is built consists of the measured terminal voltages and currents and their time derivative. In principle, one could use all the possible independent variables in an arbitrary fixed order, but this would result in models that are needlessly complex. There are more rigorous methods for selecting a subset of independent variables from which to construct the model whose output response is single-valued function [8]. The technique for finding this subset is the so-called 'false nearest neighbors' method or information theory [27].

Once the dynamic order is selected of $f_{D1}(\cdot)$ and $f_{D2}(\cdot)$, the modeling task is essentially a function approximation problem. After the parameters extraction, the model has to be synthesized as an equivalent circuit to be included in the simulators by converting the estimated discrete-time domain model into a continuous-time domain state-space realization [23]. Such an approximation is described below for the simple case.

$$i_j(t) = f_j\Big(\Theta_j,\ \big[i_j(k-1),\ v_2(k),\ v_2(k-1)\big]^{\mathrm{T}}\Big) \tag{34}$$

This relation is written as a state space realization, with the auxiliary variables $z_1(k) = i_j(k-1)$ and $z_2(k) = v_2(k-1)$.

$$\begin{cases} z_1(k) - z_1(k-1) = & f_j\big(\Theta_j, x_j(k-1)\big) - z_1(k-1) \\ z_2(k) - z_2(k-1) = & v_2(k-1) - z_2(k-1) \\ i_j(t) = & f_j\big(\Theta_j, x_j(k)\big) \end{cases} \tag{35}$$

where $x(k) = [z_1(k), v_2(k), z_2(k)]^T$. The model's discrete-time representation is restored to its continuous-time form:

$$\begin{cases} \frac{d}{dt}\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \frac{1}{T}\begin{bmatrix} f_j\left(\Theta_j, x_j(t)\right) - z_1(t) \\ v_2(t) - z_2(t) \end{bmatrix} \\ i_j(t) = f_j\left(\Theta_j, x_j(t)\right) \end{cases} \tag{36}$$

This state-space equation can be implemented as a voltage-controlled current, and charge sources or current controlled flux source in any CAD tools enabling the time derivatives in each time step of terminal voltages and currents in fitting functions to be evaluated.

6 Conclusion

The evaluation of the signal integrity is of a paramount importance with the recent trends in IC design characterized by the downscaling of the package integration and the clock frequency. As a result, the number of failures caused by SI problems is on the rise because existing modeling methodologies for I/O buffers cannot address these issues effectively. In order to overcome the TL limitations, behavioral modeling appears to be effective. Such high-level abstractions are developed to characterize the device behavior by subjecting it to various types of excitation to find out the different sources of nonlinearity and dynamics and their effects on device's behavior. This enables fast and accurate prediction of magnitude and timing waveform quality degradation due to these physical imperfections. In addition, nonlinear behavioral models can be classified in two types according to the physical knowledge reflected in the model generation's process such as the black-box and gray-box techniques.

Since the correct operation of I/O buffers is crucial to reliably transmit data through the high-speed digital link, developing an efficient model to accurately simulate their nonlinear dynamic behavior is a challenging task that is motivating several research activities. In fact, the previously neglected nonlinear dynamic effects in the active I/O buffers should be captured in the supplied model to the designer in order to perform accurate time domain simulation. Various methodologies can be followed to select the behavioral model's structure and different techniques can be used to extract its functions and parameters. Furthermore, the issues of the interaction among the design of experiment, data analysis, model generation, and validation necessary for good black-box and gray-box behavioral modeling are reviewed in this chapter.

References

1. Prodanov, W., Valle, M., Buzas, R., & Pierscinski, H. (2008, May). Behavioral models of basic mixed mode circuits: Practical issues and application. In *IEEE european conference on circuit theory and design* (pp. 854–857).
2. Caniggia, S., & Maradei, F. (2008). *Signal integrity and radiated emission of high-speed digital systems*. New York: Wiley.
3. Pen Li, M. (2007). Jitter, noise, and signal integrity at high-speed, Prentice hall modern semiconductor series.
4. Jun, F., Xiaoning, Y., Kim, J., Archambeault, B., & Orlandi, A. (2010). Signal integrity design for high-speed digital circuits: Progress and directions. *IEEE Transactions on Electromagnetic Compatibility, 52*(2), 392–400.
5. Chenjie, Gu. (2012). Challenges in post-silicon validation of high-speed I/O links. In *International conference on computer aided design*.
6. Dghais, W., Cunha, T. R., & Pedro, J. C. (2010, April). Behavioral model for high-speed digital buffer/driver. In *Integrated nonlinear microwave and millimeter-wave circuits* (pp. 110–113).
7. Gielen, G. (2005). CAD tools for embedded analogue circuits in mixed-signal integrated systems on chip. *IEEE Proceedings Computers and Digital Techniques, 152,* 317–332.
8. Wood, J., Root, D. E., & Tufillaro, N. B. (2004, September). A behavioral modeling approach to nonlinear model-order reduction for RF/microwave ICs and systems. *IEEE Transactions on Microwave Theory and Techniques, 52*(9).
9. Ljung, L. (1987). *System identification: Theory for the user*. Prentice-Hall.
10. Aaen, P., Pla, J., & Wood, J. (2007). *Modeling and characterization of RF and microwave power FETs*. New York: Cambridge University Press.
11. Maas, S. (1992). *Microwave mixers*. Norwood. MA: Artech House.
12. Varma, A. K., Steer, M., & Franzon, P. D. (2008). Improving behavioral IO buffer modeling based on IBIS. *IEEE Transaction on Advanced Packaging, 31*(4), 711–721.
13. Dang, T., Donze, A., & Maler, O. (2004). Verification of analog and mixed-signal circuits using hybrid systems techniques. In A. Hu, & A. Martin (Eds.), *Formal methods for computer aided design* (Vol. 3312, pp. 21–36). Heidelberg: LNCS, Springer.
14. Zhu, T., Steer, M. B., & Franzon, P. D. (2011, August). Accurate and scalable IO buffer macromodel based on surrogate modeling. *IEEE Transaction on Component, Packaging and Manufacturing Technology, 1*(8), 1240–1249.
15. Schreurs, D., Wood, J., Tufillaro, N., Usikov, D., Barford, L., & Root, D.E. (2000). The construction and evaluation of behavioral models for microwave devices based on time-domain large-signal measurements. In *IEEE international electron devices meeting* (pp. 819–822).
16. Pedro, J. C., & Maas, S. A. (2005, April). Comparative overview of microwave and wireless power-amplifier behavioral modeling approaches. IEEE *Transactions on Microwave Theory and Techniques,* 53(4), 1150–1163.
17. Sjöberg, J., et al. (1995). Nonlinear black-box modeling in system identification: A unified overview. *Automatica, 31*(12), 1691–1724.
18. Curras-Francos, M., Tasker, P., Fernandez-Barciela, M., Campos-Roca, Y., & Sanchez, E. (2000). Direct extraction of nonlinear FET Q-V functions from time domain large-signal measurements. *IEEE Microwave and Guided Wave Letters*, 531–533.
19. Schreurs, D., Verspecht, J., Vandenberghe, S., Carchon, G., van der Zanden, K., & Nauwelaers, B. (1999). Easy and accurate empirical transistor model parameter estimation from vectorial large-signal measurements. In *IEEE MTT symposium*.
20. Root, D. E., Wood, J., Tufillaro, N., Schreurs, D., & Pekker, A. (2002, October). Systematic behavioral modeling of nonlinear microwave/RF circuits in the time domain using techniques from nonlinear dynamical systems. In *Proceedings of the IEEE international workshop on behavioral modeling and simulation* (pp. 71–74).

21. Chan, J. Z., Allebach, J. P., & Bouman, C. A. (1997, September). Sequential linear interpolation of multidimensional function. *IEEE Transaction on Image Processing, 6*(9).
22. IBIS (I/O Buffer Information Specification)—Ver. 6.1. (2015, September). [online] Available https://ibis.org/.
23. Stievano, I. S., Canavero, F. G., & Maio, I. A. (2002). Parametric macromodels of digital I/O ports. *IEEE Transactions on Advanced Packaging, 25*(2), 255–264.
24. Dghais, W., Cunha, T. R., & Pedro, J. C. (2013, October). A novel two-port behavioral model for I/O buffer overclocking simulation. *IEEE Transactions on Components, Packaging and Manufacturing Technology, 3*(10), 1754–1763.
25. Mutnury, B., Swaminathan, M., & Libous, J. P. (2006). Macromodeling of nonlinear digital I/O drivers. *IEEE Transactions Advanced Packaging, 29*(1), 102–113.
26. Dghais, W., Cunha, T. R., & Pedro, J. C. (2012, December). Reduced-order parametric behavioral model for digital buffers/drivers with physical support. *IEEE Transactions on Components, Packaging and Manufacturing Technology, 2*(12), 1–10.
27. Sjöberg, J., et al. (1995). Nonlinear black-box modeling in system identification: A unified overview. *Automatica, 31*(12), 1691–1724.

http://www.springer.com/978-3-319-72214-6

Real-Time Modelling and Processing for Communication Systems
Applications and Practices
Alam, M.; Dghais, W.; Chen, Y. (Eds.)
2018, XI, 282 p. 210 illus., Hardcover
ISBN: 978-3-319-72214-6