Automated Nonlinear Macromodelling of Output Buffers for High-Speed Digital Applications

Ning Dong, Jaijeet Roychowdhury
Department of Electrical and Computer Engineering, University of Minnesota
Email: {ningd,jr}@ece.umn.edu

ABSTRACT

We present applications of a recently developed automated nonlinear macromodelling approach to the important problem of macromodelling high-speed output buffers/drivers. Good nonlinear macromodels of such drivers are essential for fast signal-integrity and timing analysis in high-speed digital design. Unlike traditional black-box modelling techniques, our approach extracts nonlinear macromodels of digital drivers *automatically* from SPICE-level descriptions. Thus it can naturally capture transistor-level nonlinearities in the macromodels, resulting in far more accurate signal integrity analysis, while retaining significant speedups. We demonstrate the technique by automatically extracting macromodels for two typical digital drivers. Using the macromodel, we obtain about 8× speedup in average with excellent accuracy in capturing different loading effects, crosstalk, simultaneous switching noise (SSN), *etc.*.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids—*simulation*

General Terms

Algorithms

Keywords

nonlinear macromodeling, I/O buffer macromodeling

1. INTRODUCTION

The ever-increasing speed and complexity of digital designs is often accompanied by difficult design and verification challenges. Ensuring *signal integrity* (which refers to a broad set of problems, such as crosstalk noise, electromigration, power/ground noise due to simultaneous switching outputs, *etc.*) is essential for meeting two fundamental requirements in digital design, correct timing and adequate signal quality. The traditional binary idealization of digital logic operation has long ceased to be adequate, as transistor feature sizes have shrunk and device operation has become increasingly non-ideal. This is especially true for deep-submicron technologies, in which digital logic signals are almost indistinguishable, at first sight, from continuous analog-like waveforms. Fast and reliable methodologies for assessing signal integrity are thus a key step in ensuring the quality of signals in both on- and off-chip high-speed data communications.

Simulating the interaction of I/O buffers and interconnect lines is an important signal integrity verification task. Because of the large size of today's digital systems, direct SPICE-level analog simulation has become essentially infeasible due to its computational complexity. One way of reducing this complexity has been to use macromodels of I/O buffers, *i.e.*, to replace underlying digital I/O buffers with functionally similar but computationally much simpler models.

There have been several techniques proposed for generating I/O buffer macromodels over the last decade. The most popular such method appears to be the Input/Output Buffer Information Specification (IBIS) [1]. The core of IBIS consists of lookup tables of current *vs.* voltage at the output port, together with timing information. One reason why such table models are popular amongst IC vendors is that I/O buffer model is a black box that does not reveal transistor-level details of the internal circuitry. As a result, virtually all IC vendors provide IBIS models of their products to customers, and many EDA tools support the simulation of IBIS models.

Despite its commercial success, IBIS-like modelling has intrinsic limitations. Important non-ideal effects captured, such as overshoot, undershoot, pull-up/put-down characteristics, *etc.*, have to be determined *a-priori*, largely either by measurements or by simulations for creating the I-V datasheet. 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 difficult to include any inherent nonlinearities presented in transistor-level design with cutting-edge technologies. Neither can it capture accurate higher-order dynamics of the driver as it mainly relies on the static I-V information, primarily revealing the DC characteristics.

Recently, a newer class of black-box methods (*e.g.*, [12, 19, 20]) was proposed with the goal of better capturing I/O driver dynamics. In additional to the static I-V characteristics used in IBIS models, this class of methods uses Radial Basis Functions (RBFs) or splines in order to obtain "higher-order" accuracy in capturing the output behavior of driver circuits. To create such a model, one needs to stimulate the output port of the buffer carefully 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.*, [12, 19, 20]) that these methods are capable of representing the driver circuit well and capturing the effects like crosstalk, SSO, *etc.*. However, the efficacy of black-box methods relies heavily on the choice of model representations, data sets generation and interpretation.

In the past few years, there has been progress made on alternative for *automating* nonlinear macromodel generation (*e.g.*, [3,4,14, 17, 21]). These so-called "white-box" methods start from detailed SPICE-level circuit descriptions and extract macromodels automatically via mathematical algorithms in a bottom-up fashion, with no

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

manual knowledge of underlying internal circuit structure or operation required. Compared with black-box counterparts, one of the advantages of such methods is that nonlinearities and dynamics from increasingly sophisticated device models can be naturally and automatically abstracted into the macromodels. For this reason, it has been argued that white-box methods can produce more accurate macromodels compared to black-box one. The extracted macromodels, essentially set of nonlinear differential equations of small size, can be easily represented in high-level modelling language, including MATLAB, Verilog-A, VHDL-AMS, *etc.*, and thus supported by current EDA tools.

Furthermore, white-box methods retain or even enhance a very desirable property of black-box methods, that the macromodel will "anonymize" the original circuit. It is important to note that although details of the circuit are required by the algorithm that generates the macromodel, the macromodel itself consists of abstract differential equations, from which it is essentially impossible to reverse-engineer any details of the original circuit. Hence, white-box macromodelling is also a very effective means of protecting intellectual property while retaining macromodel accuracy.

In this paper, we apply recently proposed white-box nonlinear macromodelling methods based on piecewise representatiosn (*i.e.*, TPWL [14], PWP [3]), to automatically extract I/O buffer macromodels. One advantage of our approach that it produces *a single macromodel* that can be used as a drop-in replacement for a variety of different type of simulations. The essence of piecewise macromodelling is to explore relevant sections of internal state-space of the circuit, splitting them up into piecewise regions, within which a reduced-order polynomial model is used to approximate the original nonlinearities, and stitching them together using a smoothing function.

When extracting models of high-speed I/O buffers, it is particularly important to capture the effects of different resistive, inductive and transmission-line *loading* well, in addition to clipping/saturation and nonlinear slewing. Because these effects are fundamentally both nonlinear and dynamical, they constitute a special challenge to macromodel well. We show in this paper that TPWL and PWP are more than adequate for this task. By employing *multiple training inputs* during generation of the macromodel to improve state-space coverage, and by using novel, smooth *weight functions* for stitching together the piecewise regions, we obtain accurate results for predicting logic level bounce (SSN) and crosstalk while obtaining speedups of about an order of magnitude for two typical I/O buffers. Indeed, because capturing weakly nonlinear phenomena with high dynamic range is not usually critical for I/O buffer applications, we find that using only piecewise linear (PWL) macromodels, with weight function and training enhancements, is perfectly adequate. Note that the weight functions and training procedures we use distinguish our approach from the TPWL method [14] as originally proposed.

The remainder of the paper is organized as follows. In Section 2, we review existing black-box approaches for I/O buffer macromodelling as well as some background for piecewise macromodelling, discussed in Section 3. We then apply piecewise macromodelling to two I/O buffer examples to generate macromodels, and compare SSN/crosstalk simulation speeds and accuracies with full SPICE-level circuits, in Section 4.

2. PREVIOUS WORK AND BACKGROUND

2.1 Black-box Macromodelling for I/O Buffers

Black-box methods refer to a broad set of modelling approaches, such as artificial neural networks (ANN) (*e.g.*, [23]), radial basis functions (RBF) (*e.g.*, [12, 19]), lookup tables (*e.g.*, [22]), *etc.*. They treat the system being modelled as a black box and reverse-engineer input-output behavior using data sampled from simulation or measurement. Precise details of internal structure are not required; however, making good assumptions of functionality is important.

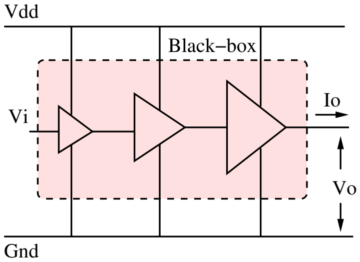


Figure 1: Black-box model of digital buffer

Fig. 1 shows a 2-port black-box model of an output buffer, whose dynamics can be estimated by the data set of I-V characteristics through the output port, *i.e.*, $\{V_{o_i}, I_{o_i}\}$, $i = 1, 2, \ldots, N$. Using these as "training data", a black-box model can be created by finding a proper function representation f, such that

$$I_{o_i} = f(V_{o_i}), i = 1, 2, \ldots, N.$$

The model $I_o = f(V_o)$ is then applied to generic practical data either within or outside the training set.

Many functional representations have been proposed for modelling dynamic systems in black-box form. For example, when discrete time-sampled data is avaialble, a multiple-input single-output parametric model that has been widely used to model system dynamics in automatic control can be written as

$$y(k) = F(x(k), \Theta),$$

where $x(k) = [y(k-1), \ldots, y(k-r), u(k), \ldots, u(k-r)]^T$ is a regressor vector that consists of the past r samples of input sequences $u(k)$ and output sequences $y(k)$. $F : \mathbb{R}^{2r+1} \rightarrow \mathbb{R}$ is a nonlinear function, Θ consists of all model parameters that will be determined by "curve-fitting" using those training data, r is the dynamic order of the model, representing the "memory" feature of the system.

There are many ways of choosing F. As illustrated in [20], one way is to use Gaussian radial basis functions (RBFs),

$$F(x(k), \Theta) = \sum_{j=1}^{p} \alpha_j \Phi(||x(k) - c_j||, \beta_j),$$

where Φ is a scalar generator function, *i.e.*, $\Phi(\xi, \beta) = e^{-\frac{\xi^2}{\beta^2}}$ generating all the basis functions with different β and ξ. β_j, ξ_j and c_j are modelling parameters and are encapsulated in Θ. Sigmoidal (SIG) functions for I/O buffer macromodelling have been investigated in [19]. To determine the parameters Θ, various "curve-fitting" technologies, such as least-square, nonlinear regression, *etc.*, have been employed. By carefully choosing sampling data to best reveal dynamics of the buffer, the generated black-box macromodel can produce very good results for signal integrity analysis, with up to 40× speedups [19, 20].

However, with driver circuits becoming more and more complex in modern digital systems, it is questionable if modelling internal dynamics only by fitting the input-output data will remain an effective and sustainable approach. Since internal circuit details are generally available for circuits (as SPICE netlists which are extensively used during design), a promising option is to apply CAD tools for push-button generation of nonlinear macromodels from their SPICE-level descriptions. In this paper, we explore the application of the recently developed TPWL [14] and PWP techniques [3] for extracting nonlinear macromodels of I/O buffers automatically.

2.2 Background

TPWL [14] and PWP [3] are for macromodelling general nonlinear circuits and systems that can be described by differential algebraic equations (DAEs) as

$$\dot{q}(x(t)) = f(x(t)) + b(t). \tag{1}$$

All variables (except time t) are vector-valued. $x(t) \in \mathbb{R}^n$ are the n unknown variables (including node voltages and branch currents in circuits); q denotes the charge/flux terms and f the resistive terms; $b(t)$ is the vector of excitations to the circuit. Without loss of generality, (1) can be expressed as (*e.g.*, [16])

$$E\dot{x} = f(x) + Bu(t), \quad y = Cx, \tag{2}$$

where $E \in \mathbb{R}^{n\times n}$. m-inputs $u(t) \in \mathbb{R}^m$ and p-outputs $y(t) \in \mathbb{R}^p$ are connected to internal states by matrices $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$ respectively.

The most commonly used technique in model-order reduction (MOR) (including TPWL/PWP) is Krylov-subspace projection, most simply explained for linear time-invariant (LTI) systems. First linearize the nonlinear system (2) at some point, such that,

$$E\dot{x} = Ax(t) + Bu(t), \quad y(t) = Cx(t), \tag{3}$$

where the matrices $A \in \mathbb{R}^{n\times n}$ is the Jacobian matrix. The transfer function of (3) and its Taylor series expansion at $s = 0$ are given by

$$H(s) = C(sE - A)^{-1}B, \tag{4}$$

$$H(s) = -C[A^{-1}B + s(A^{-1}E)A^{-1}B + s^2(A^{-1}E)^2A^{-1}B + \ldots].$$

The coefficients of s, such as $CA^{-1}B$, $C(A^{-1}E)A^{-1}B$, *etc.*, are usually called *moments*.

Krylov-subspace techniques provide a projection matrix $V \in \mathbb{R}^{n\times q}$ ($q \ll n$), whose columns span a *Krylov subspace* defined as

$$K_q = span\{A^{-1}B, (A^{-1}E)A^{-1}B, \ldots, (A^{-1}E)^{q-1}A^{-1}B\}. \tag{5}$$

After projecting the original state-space $x \in \mathbb{R}^n$ into this subspace via $x = Vz$ (*e.g.*, [6–9]), the time-domain state-space representation of (3) will be reduced to

$$\hat{E}\frac{dz}{dt} = \hat{A}z(t) + \hat{B}u(t), \quad y = \hat{C}z(t),$$

where the reduced matrices are $\hat{E} = V^TEV \in \mathbb{R}^{q\times q}$, $\hat{A} = V^TAV \in \mathbb{R}^{q\times q}$, $\hat{B} = V^TB \in \mathbb{R}^{q\times m}$, and $\hat{C} = CV \in \mathbb{R}^{p\times q}$. $z(t) \in \mathbb{R}^q$ are the states of reduced system. The *reduced* transfer function is

$$\hat{H}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B}. \tag{6}$$

The projection bases V can be calculated via, *e.g.*, the Lanzcos or Arnoldi methods([5, 6]). It has been proved (*e.g.*, [6, 7]) that the first q moments of (6) match the first q moments of (4), implying a good approximation to the original LTI system.

This Krylov-subspace technique can be extended to weakly nonlinear system, for which the original system of (2) has small-signal polynomial expansion around some operating point,

$$E\dot{x} = A_1x + A_2x \otimes x + A_3x \otimes x \otimes x + \cdots + Bu(t),$$

where A_i is the ith order derivative and the symbol $\otimes$ stands for Kronecker tensor product. Similarly as LTI system, a Krylov-subspace projection basis V can also be calculated. The key difference from LTI system, however, is that the projection basis V should consist of not only the linear information, but more important the higher-order derivative information in order to preserve effective nonlinearities in the reduced model. The detail algorithms of finding such a basis are referred to, *e.g.*, [11, 13, 15].

3. NONLINEAR MACROMODELLING

For general nonlinear systems, with strong nonlinearities, a trajectory piecewise linear (TPWL) approach was first proposed in [14] and then extended to PWP [3]. The basic approach is to partition the internal state-space into piecewise regions, each approximated by a linear or polynomial model of smaller size. Each region is reduced in size, and the reduced piecewise models are then stitched together into one macromodel using a smoothing function. The TPWL/PWP formulations and important implementation details are recapitulated in this section.

3.1 Piecewise Polynomial Representations

With certain input $u(t)$, the solution $x(t)$ to general nonlinear system of (2) can be viewed as a trajectory in $\mathbb{R}^n$. We choose s expansion points $\{x_1, x_2, \ldots, x_s\}$ ($x_i = x(t_i)$) along this trajectory, for each of which a small signal polynomial expansion is given by

$$E\dot{x} = f(x_i) + A_i^{(1)}x^{(1)} + A_i^{(2)}x^{(2)} + Bu(t), \quad y = Cx.$$

Here $x^{(1)} = x - x_i$, $x^{(2)} = (x - x_i) \otimes (x - x_i)$ is the Kronecker tensor product, $A_i^{(1)}$ and $A_i^{(2)}$ are first and second derivatives (For simplicity, we only expand systems up to quadratic term. Extension to higher order terms is straightforward).

Next, we generate projection bases V_i for each polynomial model with polynomial MOR techniques[1]. Similarly as TPWL [14], a uniform projection base V is constructed via singular value decomposition (SVD) on the collections of all bases. The final size of V is properly chosen by examining the singular values such that V will contain the common features among all project bases. If $V \in \mathbb{R}^{n\times q}$ ($q < n$), each polynomial will be reduced to size q, *i.e.*,

$$\hat{E}\dot{z} = \hat{f}(x_i) + \hat{A}_i^{(1)}z^{(1)} + \hat{A}_i^{(2)}z^{(2)} + \hat{B}u(t),$$

where $z_i = V^Tx_i$, $z^{(1)} = z - z_i$, $z^{(2)} = (z - z_i) \otimes (z - z_i)$, $\hat{f}(x_i) = V^Tf(x_i)$. The reduced matrices $\hat{E} = V^TEV$, $\hat{B} = V^TB$, $\hat{A}_i^{(1)} = VA_i^{(1)}V$ and$\hat{A}_i^{(2)} = V^TA_i^{(2)}V \otimes V$.

The final model is obtained by a weighted combination of these polynomial models, such that

$$\begin{aligned} \hat{E}\dot{z} &= \sum_{i=1}^{m} w_i(z)(\hat{f}(x_i) + \hat{A}_i^{(1)}z^{(1)} + \hat{A}_i^{(2)}z^{(2)} + \hat{B}_iu(t)), \\ y &= C[\sum_{i=1}^{m} w_i(z)(x_i + V(z - z_i)], \end{aligned}$$

where $w_i(z)$ is a scalar weight function that is elaborated in Section 3.2.3.

3.2 Implementation Details

3.2.1 Merge regions from multiple trainings

Intuitively, one training input and the associated solution (trajectory) can only cover a limited range of the state-space. The key to generating widely-applicable models is to increase state-space coverage by merging piecewise regions from multiple training trajectories. As illustrated in Fig. 2, once the system trajectory falls within the covered range, good approximations can be expected from the macromodel. Once the state is out of the model's range of validity and large errors occur, new regions should be inserted into the piecewise model. Ideally, one should incrementally refine the macromodel whenever new regions are necessary, to ensure accuracy. In this paper, training trajectories are obtained through several transient simulations of the output buffer, with different loads.

[1] Any weakly polynomial MOR technique can be used to generate the projection basis, *e.g.*, [11, 13, 15]. If only expanded to linear model, one can use normal Krylov-subspace as in (5).

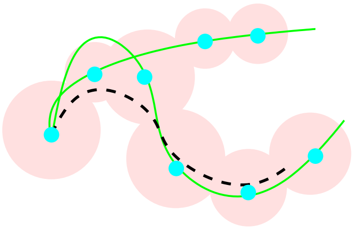


Figure 2: Merging expansion regions with multiple training trajectories (solid line: training trajectories; dash line: trajectory in practical simulations)

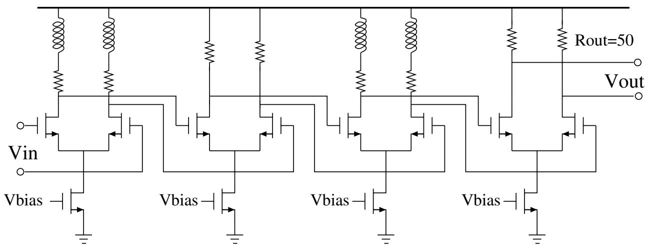


Figure 3: Tapered current-model logic (CML) buffer

3.2.2 *Constructing uniform projection basis*

MOR for each region is carried out through the projection $x = V_i z$, which projects $x \in \mathbb{R}^n$ from original space into the reduced space $z \in \mathbb{R}^{q_i}$. z is the local coordinate of x corresponding to subspace V_i. Different reduced polynomial models thus have different local coordinate systems. To stitch them together, one approach is to find *one* common subspace (coordinate system), possibly larger but encapsulating all the underlying (smaller) subspaces. A straightforward solution is to collect dominant information using SVD, *i.e.*, $V = svd([V_1, V_2, \ldots, V_s])$ and keep only q ($q < n$) vectors according to first q leading singular values. The key observation here is that the singular value will suddenly drop to a very small quantity at certain position, implying the existence of such a common subspace.

3.2.3 *Weight Function*

A scalar weight function [3, 14] is used to smooth transitions when crossing region boundaries. The value of the weight function $w_i(z)$ should be close to 1 when the state vector z approaches the center point z_i, and should attenuate to zero rapidly as z leaves z_i.

The weight function is critically important for transient simulation with large signal inputs. Intuitively, the dynamics within each region are governed by the polynomial model inside; for transitions near the boundaries, they rely on the weight function to choose the proper region and smooth the trajectory out. **If the weight function is not well defined, or if its derivative is not continuous, large-signal transient simulation typically suffers serious convergence and stability problems.**

Although there is considerable flexibility in the choice in functions for this purpose, it is not trivial to devise good weight functions that are smooth and differentiable. After much experimentation, we have developed the following weight function (for PWP), which has worked satisfactorily for a variety of applications:

$$w_i(z) = [\frac{d_{\min}(z)}{d_i(z)} e^{-\frac{d_i(z)-d_{\min}(z)}{D_{\min}}}]^p, \qquad (7)$$

where $d_i(z) = ||z - z_i||_2^2$, $d_{\min}(z) = \min(d_i(z))$ for $i = 1, \ldots, s$ and $D_{\min}$ is the minimum distance among those center points $\{z_i\}$. **Note that $d_{\min}(z)$ is differentiable, in spite of the fact that the** $\min(\cdot)$ **function itself is not.** The parameter p (typically $p = 1 \sim 2$) is used to make transition near the boundary smoother or sharper. The weight function is finally normalized to satisfy $\sum_{i=1}^{s} w_i(z) = 1$.

4. EXPERIMENTAL RESULTS

In this section, we apply the macromodelling procedures described in Section 3 to generate macromodels for two output buffer structures commonly used in high-speed digital design.

The first example, as shown in Fig. 3, is a tapered current-mode logic (CML) buffer chain [18] ($V_{dd} = 1.8V$). The resistor in the last stage is set to match 50Ω transmission line. Inductive peaking is employed in the first and the third stage to increase the bandwidth. The sizing of each stage and parameters are optimized to minimize the buffer delay [10]. The circuit size of this example is 28.

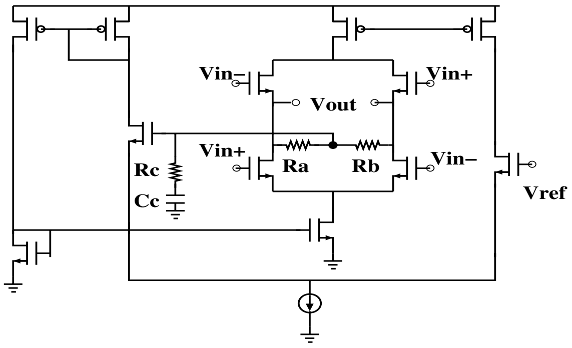


Figure 4: Low-voltage differential signaling (LVDS) buffer with common-mode feedback loop

The second example is a low-voltage differential signaling (LVDS) output buffer ($V_{dd} = 3.3V$) with common-mode feedback loop [2] as shown in Fig. 4. The common-mode voltage of inputs is enforced by V_{ref} to be around $1.25V$. It is designed to drive 50Ω with about 0.5V voltage swing. The size of the circuit is 18.

4.1 Macromodel Generation

For digital applications, we are primarily interested in the switching activities of the buffers with large signal input, which are dominated by the coverage of piecewise regions and the smoothing function. For such cases, weak nonlinearities captured by the polynomials inside each region are not as important as in other applications such as op-amp and mixer macromodelling. Through experimentation, we have found that using linear-only models within each region is adequate for meeting accuracy requirements[2]. Importantly, the weight function (7), together with merging multiple training trajectories as dsecribed in Section 3.2, are both very important for developing macromodels that work well in large signal transient analysis.

Fig. 5 illustrates the block diagram for macromodel generation. The buffer is modelled with 5 inputs and two outputs: two differential inputs track different input patterns; two loading currents tackle loading variations; power grid noise is captured via port V_s. Two differential outputs are connected to the load.

Several transient simulations of the full buffer circuit with input pattern “010” and different loads (*e.g.*, 50Ω resistor and $1pF$ capacitor) are used to generate the training trajectories, along which the piecewise regions are selected and merged. All circuit simulations and verifications are based on the MATLAB/Linux platform using modified Schichman-hodges model.

[2] Being able to leave out polynomial terms improves the macromodel's efficiency significantly.

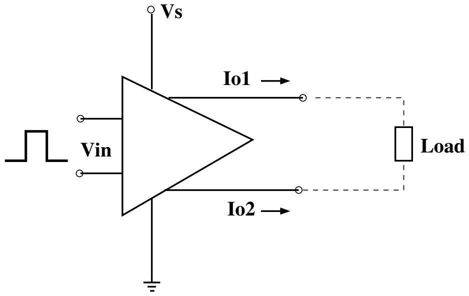


Figure 5: Macromodel generation and input/output ports

The total generation time is about 610s for the CML buffer and ∼400s for the LVDS example. Over 90% of this time is spent on full system simulations. The final PWL macromodel for the first CML buffer consists of 32 regions with reduced size of 15 (originally 28). The macromodel for the LVDS example has 21 regions, each of which the size has been reduced from 18 to 11. Finally, they are stitched together smoothly using the weight function (7).

4.2 Different Loading Effects

We verify capturing different loading effects using the macromodel from the first example (CML buffer). Three transmission lines (modelled with lumped RLC network) are connected to the buffer in the test. The voltage waveforms across the load at far-end of the transmission line against full circuit simulation are shown in Fig. 6. The three cases are:

(a) lossless transmission line, $Z_c = 75\Omega$, $T_d = 0.4ns$, $Z_{load} = Z_c$, input pattern “0100101”;

(b) lossy transmission line, $Z_c = 100\Omega$, $T_d = 0.5ns$, $Z_{dc} = 2\Omega$, $Z_{load} = Z_c$, input pattern “0101100”;

(c) lossy transmission line, $Z_c = 75\Omega$, $T_d = 0.5ns$, $Z_{dc} = 2\Omega$, $Z_{load} = 1pF$, input pattern “0110010”.

It is seen that the macromodel is capable of capturing different loading effects and its accuracy in matching the full circuit simulation is more than adequate. The relative error is less than 5% on average.

The typical runtime of transient simulation, *e.g.*, case (c) is about 885.82s (full system) vs 112.3s (macromodel), representing about 8× speedup.

4.3 Crosstalk

We further investigate the CML buffer macromodel for crosstalk simulation. As illustrated in Fig. 7, two coupled lossy transmission lines ($Z_c = 75\Omega$, $T_d = 0.5ns$, $Z_{dc} = 2\Omega$) are driven by two buffers: one is active with input pattern “0101100” and the other remains quiet.

The voltages waveform on the load impedance at far-end of both lines are shown in Fig. 8. It is seen that the macromodel reproduces the dynamic behaviors of the buffer and capture the crosstalk noise quite well.

For this test, about 8.8× speedup resulted from using macromodel (138.1s) against full system simulation (1215.9s). The relative error is 6.7% on average.

4.4 Simultaneous Switching Noise (SSN)

The macromodel of the second example (LVDS buffer in Fig. 4) is used in this test. As shown in Fig. 9, M identical drivers are loaded with lossy transmission line ($Z_c = 100$, $T_d = 0.5ns$, $Z_{dc} = 2\Omega$). An ideal power supply V_{dd} was connected to the power supply port V_s of drivers through L_s and R_s. In the simulation, $M = 7$, $L_s = 0.1nH$ and $R_s = 1m\Omega$. All drivers have the same input stream “0100101”.

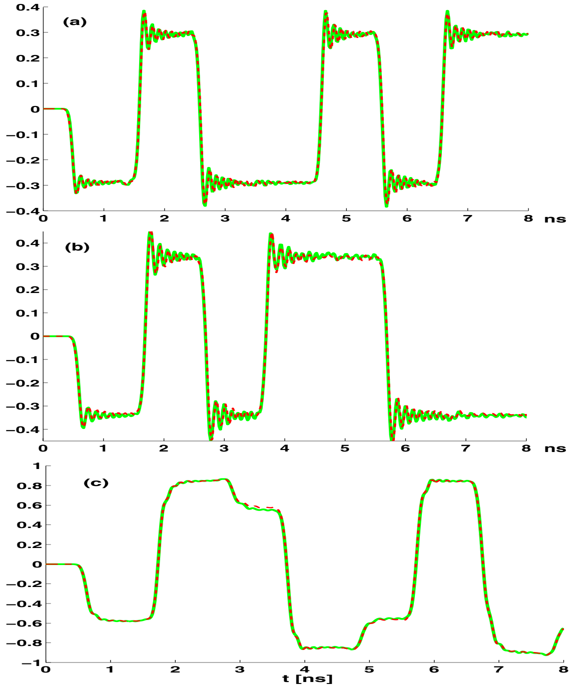


Figure 6: Voltage waveform across the load. Solid line: full circuit simulation; dash line: macromodel simulation.

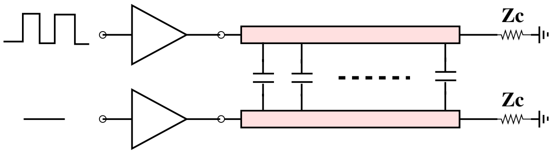


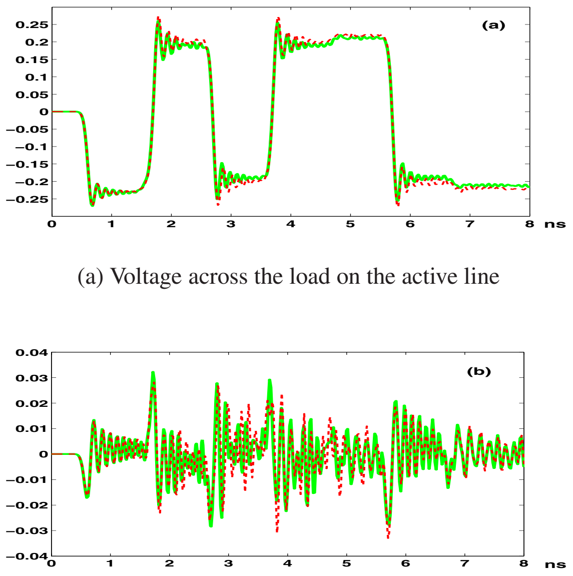
Figure 7: Circuit for crosstalk simulation

The simulation result of macromodel against full circuit for capturing the noise in node V_s and supply current I_{dd} are shown in Fig. 10.

It is seen that the macromodel accurately captures the sensitive SSN noise in both voltage and current waveforms. For this test, about 7.6× speedup was obtained, with 612.7s for full simulation vs 80.2s for the macromodel. The degraded speedup is partly because of the originally small and relatively simple LVDS circuit. In general, more significant speedups are obtained for large examples using complex device models.

5. CONCLUSIONS

In this paper, we have applied automated white-box piecewise-macromodelling methods to I/O buffers. Our approach automatically extracts the nonlinear macromodel in a methodological fashion from SPICE-level descriptions and anonymizes the original circuits for IP protection. We have shown that generated macromodels can be used as drop-in replacements in signal-integrity simulations to capture different loading effects, SSN and crosstalk noise, *etc.*. Our initial results support the expectation that, with further research and development, such automated macromodelling techniques will become the method of choice for generating high-fidelity macromodels of high-speed I/O buffers.



(a) Voltage across the load on the active line

(b) Voltage across the load on the quite line

Figure 8: Macromodel in crosstalk simulation. Solid line: full circuit; dash line: macromodel.

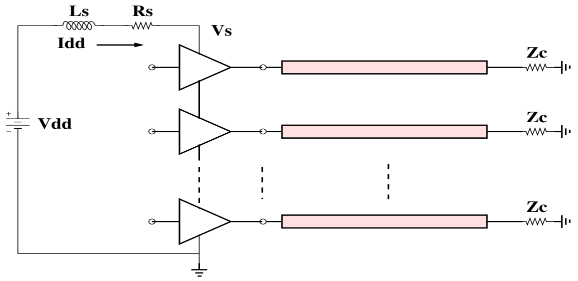


Figure 9: Simultaneous switching Noise (SSN) validation

6. REFERENCES

[1] I/O Buffer Information Specification (IBIS) Ver. 4.1. http://www.eigroup.org/ibis.

[2] A. Boni, A. Pierazzi, and D. Vecchi. LVDS I/O Interface for Gb/s-per-Pin Operation in 0.35-um CMOS. *IEEE J. Solid-State Ckts.*, 36(4):706–711, Apr. 2001.

[3] N. Dong and J. Roychowdhury. Piecewise Polynomial Nonlinear Model Reduction. *Proc. IEEE DAC*, 2003.

[4] N. Dong and J. Roychowdhury. Automated Extraction of Broadly Applicable Nonlinear Analog Macromodels from SPICE-level Descriptions. *Proc. IEEE CICC*, 2004.

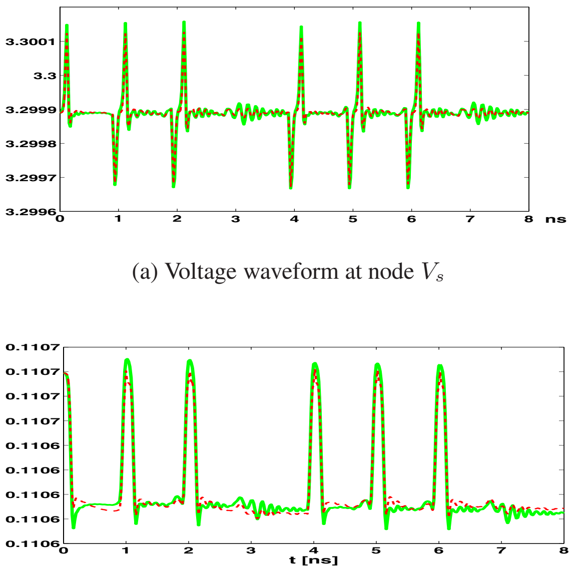
[5] P. Feldmann and R. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. CAD*, 14(5):639–649, May 1995.

[6] R. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. Technical Report 11273-980217-02TM, Bell Laboratories, 1998.

[7] R. W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *Journal of Computational and Applied Mathematics*, 123:395–421, 2000.

[8] K. Gallivan, E. Grimme, and P. V. Dooren. Asymptotic waveform evaluation via a lanczos method. *Appl. Math. Lett.*, 7:75–80, 1994.

[9] E. Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, University of Illinois, EE Dept, Urbana-Champaign, 1997.



(a) Voltage waveform at node V_s

(b) Noisy supply current $I_d d$

Figure 10: SSN using macromodel of LVDS buffer. Solid line: full circuit; dash line: macromodel.

[10] P. Heydari. Design and analysis of low-voltage current-mode logic buffers. *Fourth International Symposium on Quality Electronic Design*, pages 293–298, 2003.

[11] P. Li and L. T. Pileggi. NORM: Compact Model Order Reduction of Weakly Nonlinear Systems. *Proc. IEEE DAC*, 2003.

[12] B. Mutnury, M. Swaminathan, and J. Libous. Macro-modeling of non-linear I/O drivers using spline functions and finite time difference approximation. *Proc. Electrical Performance of Electronic Packaging*, pages 273–276, 2003.

[13] J. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Proc. IEEE DAC*, June 2000.

[14] M. Rewienski and J. White. A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices. *IEEE Trans. CAD*, 22(2), Feb. 2003.

[15] J. Roychowdhury. Reduced-order modelling of linear time-varying systems. In *Proc. ICCAD*, Nov. 1998.

[16] J. Roychowdhury. Reduced-order modelling of time-varying systems. *IEEE Trans. Ckts. Syst. – II: Sig. Proc.*, 46(10), Nov. 1999.

[17] J. Roychowdhury. An Overview of Automated Macromodelling Techniques for Mixed-Signal Systems. *Proc. IEEE CICC*, 2004.

[18] J. Savoj and B. Razavi. A 10-Gb/s CMOS clock and data recovery circuit with a half-rate linear phase detector. *IEEE J. Solid-State Ckts.*, 36:761–768, May 2001.

[19] I. Stievano, F. Canavero, Z. Chen, G. katopis, and I. Maio. Parametric macromodels of drivers for SSN simulations. *IEEE Trans. Electromagnetic Compat.*, 2:616–621, Aug. 2003.

[20] I. Stievano, I. Maio, and F. Canavero. Parametric Macromodels of Digital I/O Ports. *IEEE trans. on advanced packaging*, 25(2):255–264, May 2002.

[21] D. Vasilyev, M. Rewienski, and J. White. A TBR-based Trajectory Piecewise-Linear Algorithm for Generating Accurate Low-order Models for Nonlinear Analog Circuits and MEMS. *Proc. IEEE DAC*, 2003.

[22] B. Yang and B. MacGaughy. An Essentially Non-Oscillatory (ENO) High-Order Accurate Adaptive Table Model for Device Modeling. *Proc. IEEE DAC*, 2004.

[23] Q. Zhang and K. Gupta. *Neural Networks for RF and Microwave Design*. Boston: Artech House, 2000.