Macromodeling of Nonlinear Digital I/O Drivers

Bhyrav Mutnury, *Student Member, IEEE*, Madhavan Swaminathan, *Senior Member, IEEE*, and James P. Libous, *Senior Member, IEEE*

***Abstract*—In this paper, a modeling technique using spline functions with finite time difference approximation is discussed for modeling moderately nonlinear digital input/output (I/O) drivers. This method takes into account both the static and the dynamic memory characteristics of the driver during modeling. Spline function with finite time difference approximation includes the previous time instances of the driver output voltage/current to capture the output dynamic characteristics of digital drivers accurately. In this paper, the speed and the accuracy of the proposed method is analyzed and compared with the radial basis function (RBF) modeling technique, for modeling different test cases. For power supply noise analysis, the proposed method has been extended to multiple ports by taking the previous time instances of the power supply voltage/current into account. The method discussed can be used to capture sensitive effects like simultaneous switching noise (SSN) and cross talk accurately when multiple drivers are switching simultaneously. A comparison study between the presented method and the transistor level driver models indicate a computational speed-up in the range of 10–40 with an error of less than 5%. For highly nonlinear drivers, a method based on recurrent artificial neural networks (RNN) is discussed.**

***Index Terms*—Black-box model, finite time difference, Input/output buffer information specification (IBIS) models, Radial Basis Function (RBF), Recurrent Neural Network (RNN), Simultaneous switching noise (SSN), Spline function.**

I. Introduction

SIGNAL integrity (SI) and timing analysis of large digital systems in today's world is becoming more and more complex both in terms of CPU memory required and time consumed for simulation. One method to reduce this complexity is to use macromodels of the subcircuits constituting these large digital systems. Digital integrated circuits or digital drivers play an important role in these large digital systems and, therefore, accurate modeling of the digital drivers to capture their nonlinearity is a big challenge. This is important for efficient SI and timing analysis.

Black-box modeling techniques are helpful in modeling digital driver circuits since they are independent of the knowledge of the internal logic of the driver. Fig. 1 shows a two-port black-box circuit model of a digital buffer with information about its input and output voltage–current characteristics and no information about its internal circuitry. There have been a few techniques proposed in the past for modeling digital driver circuits [1], [2]. The most popular among them is the Input/output buffer information specification (IBIS) model. IBIS has large set of libraries, offers high computational efficiency, and has commercial software tools to handle complex modeling problems. Even though IBIS is commercially popular, it has many limitations. The fundamental limitation of IBIS is that the physical effects to be considered are decided *a-priori* when the equivalent circuit is defined, leaving little or no possibility for including the effects inherent to the device. IBIS models also fail to capture the dynamic characteristics of the driver accurately as the modeling technique relies primarily on static characteristics [3].

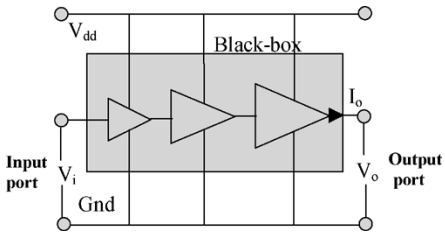


Fig. 1. Two-port black-box model of a digital driver.

Another popular modeling method is the use of radial basis function (RBF) for representing the driver input/output behavior. RBF models approximate driver circuits accurately and have been applied to complex circuits with multiple ports [4]–[6]. However, RBF modeling method has limitations due to the localized behavior of these functions. As the number of basis functions increase, RBF models become complex, and this is particularly true for complex driver circuits [7]. The complexity of the RBF parameter estimation increases as the modeling technique is extended to multiple ports. This creates two problems: 1) nonconvergence in SPICE simulations due to the ill-conditioned matrix and 2) a simulation result that is slower in computational speed than the transistor level model.

In this paper, an alternate approach using spline functions with finite time difference approximation is proposed for accurately modeling moderately nonlinear digital driver circuits. This method takes into account both the static and the dynamic characteristics of the drivers such as the current–voltage relationship and memory effects. To estimate power supply noise and cross-talk accurately, the method has been extended to multiple ports. However, for highly nonlinear circuits, this method has problems with accuracy. To alleviate this problem, a method based on recurrent artificial neural networks (RNN) is proposed and briefly discussed.

This paper is organized as follows. In Section II, spline function with finite time difference approximation modeling technique is explained in detail. Section III explains modeling of preemphasis driver circuits. In Section IV, extension of the

Manuscript received March 1, 2004; revised September 9, 2004.

B. Mutnury and M. Swaminathan are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: gt0715b@prism.gatech.edu; madhavan.swaminathan@ece.gatech.edu).

J. P. Libous is with the IBM Technology Group, Endicott, NY 13760 USA (e-mail: libousjp@us.ibm.com).

Digital Object Identifier 10.1109/TADVP.2005.848396

above modeling method to multiple ports is explained in detail. SPICE netlist generation is described in Section V. A brief description on RBF modeling technique is provided in Section VI. Different test cases that validate the accuracy and speed of this method are discussed in Section VII along with a speed versus accuracy comparison between the methods presented and actual transistor level driver models. Limitations of spline function with finite time difference approximation technique in modeling highly nonlinear driver circuits with RNN modeling technique as probable solution is discussed in Section VIII. Section IX summarizes the work with a conclusion.

II. Spline Function With Finite Time Difference Approximation

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

$$f_{n,s}(k) = A_{n,m}{v_o}^m(k) + A_{n,(m-1)}{v_o}^{(m-1)}(k) + \cdots + A_{n,0} \qquad n = 1,2; m \geq 1 \quad (1)$$

where, As are constants, v_o is the driver output voltage, and the value of m is usually less than 5.

The static submodels $f_{1,s}$ and $f_{2,s}$ can be estimated by conducting a dc sweep at the output of the driver for both driver inputs HIGH and LOW, respectively. Fig. 2(a) and (b) show the dc variation of the driver output current for different dc output voltage values for an IBM driver (AGPV3V2) when the driver input is held HIGH and LOW, respectively. It can be seen that Fig. 2(a) and (b) can be expressed as static relations using (1). Static characteristic models do not, however, capture the nonlinearity associated with the dynamic response of the driver. As the driver excitations become faster and faster, static characteristic submodels deviate from the original response. This is due to the fact that dynamic characteristics of the driver become more predominant as the excitations become faster. Fig. 3(a) shows a piecewise-linear (PWL) voltage source connected at the output of an IBM driver (AGPV3V2) when the driver input is held HIGH. The accuracy of the static model can be determined from Fig. 3(b), where it can be seen that the static model deviates from the original current response generated from the PWL voltage source connected at the output of the driver. Since the deviation is the result of failure in capturing the dynamic characteristics, static modeling in (1) can be modified to include the previous time instances of the driver output current so that dynamic behavior of the driver can be captured [7].

When the driver input is set HIGH, the current at the output i_{oh} can be expressed as submodel $f_{1,s}$ as shown in

$$f_{1,s}(k) = i_{oh}(k) = A_{1,m}{v_o}^m(k) + A_{1,(m-1)}{v_o}^{(m-1)}(k) + \cdots + A_{1,0} \quad (2)$$

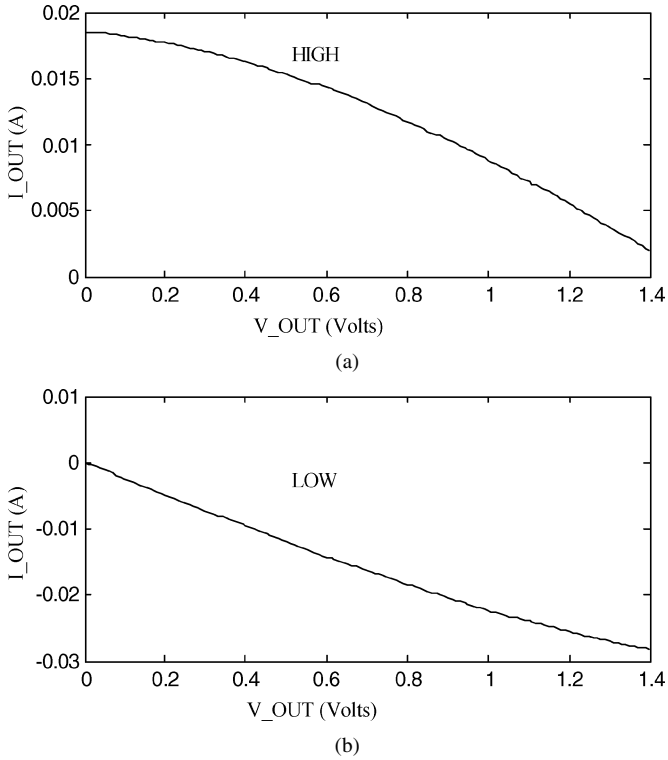


Fig. 2. (a) Static relation between an IBM driver output current and output voltage when the input is HIGH. (b) Static relation between an IBM driver output current and output voltage when the input is LOW.

Submodel $f_{1,s}$ at time instance $k-1$ can be expressed as (3)

$$f_{1,s}(k-1) = i_{oh}(k-1) = A_{1,m}{v_o}^m(k-1) + A_{1,(m-1)}{v_o}^{(m-1)}(k-1) + \cdots + A_{1,0}. \quad (3)$$

Incremental change in the driver output current Δi_{oh} is the difference between the present instance (k) and previous time instance ($k-1$) values of submodel $f_{1,s}$ as shown in (4) and (5)

$$f_{1,s}(k) - f_{1,s}(k-1) = i_{oh}(k) - i_{oh}(k-1) = \Delta i_{oh} \quad (4)$$

(Or)

$$f_{1,s}(t) - f_{1,s}(t-\Delta t) = \Delta i_{oh}. \quad (5)$$

Once Δi_{oh} is calculated, first derivative of driver output current i'_{oh} can be approximated using (6) as

$$\frac{f_{1,s}(t) - f_{1,s}(t-\Delta t)}{\Delta t} = \frac{\Delta i_{oh}}{\Delta t} = i'_{oh} \quad (6)$$

where, Δt is the sampling time. The effect of dynamic behavior when the driver input is HIGH is captured in i'_{oh}, similarly the effect of dynamic behavior when the driver input is LOW is captured by $\Delta i'_{ol}$. Therefore, dynamic behavior can be added to static submodels $f_{1,s}$ and $f_{2,s}$ as shown in (7)

$$f_1(k) = f_{1,s}(k) + p^* i'_{oh}$$
$$f_2(k) = f_{2,s}(k) + pp^* i'_{ol} \quad (7)$$

where p and pp are constants whose magnitude can be estimated by calculating the error between $f_{1,s}/f_{2,s}$ and transistor level driver output current values for inputs HIGH/LOW, respectively. It is important to note that there is no limitation on the number

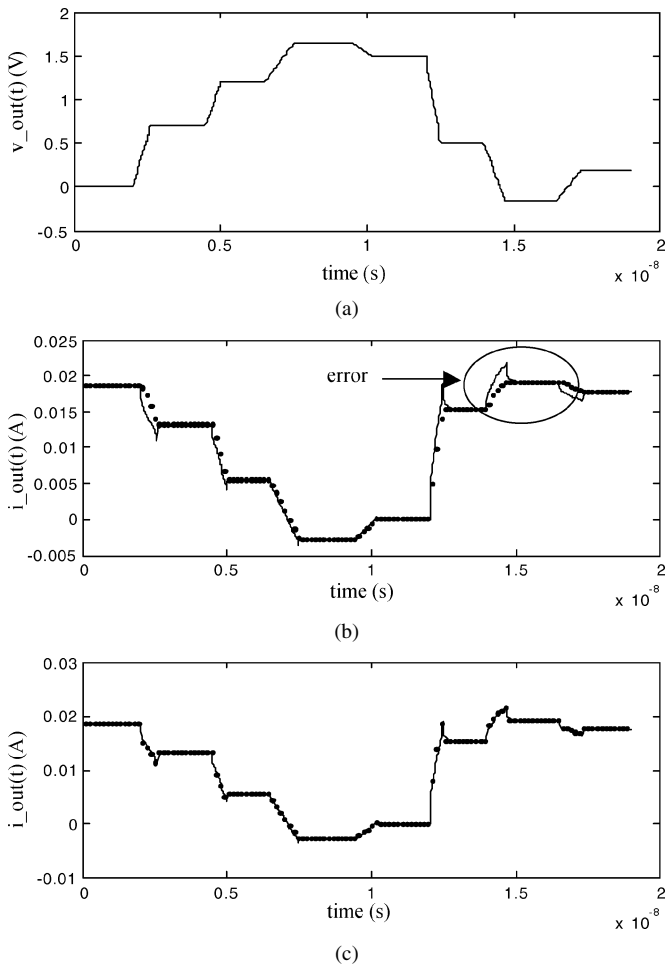


Fig. 3. (a) PWL voltage source connected at the driver output for input HIGH. (b) Output current from an IBM transistor level model (straight line) and from static characteristics model (dotted line). (c) Output current from an IBM transistor level model (straight line) and from static characteristics with finite time difference approximation model (dotted line).

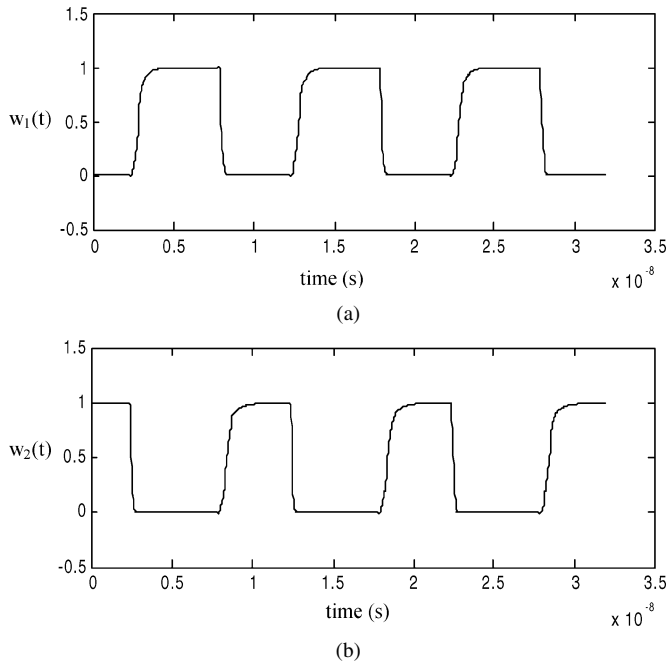


Fig. 4. (a) Weighting function w_1 that helps submodel f_1 to transit from HIGH to LOW. (b) Weighting function w_2 that helps submodel f_2 to transit from LOW to HIGH.

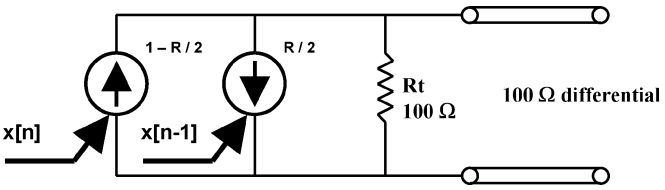


Fig. 5. A two-tap FIR driver pre-compensation scheme.

of previous output current time instances that can be added to the static submodels $f_{1,s}/f_{2,s}$ as shown in (8). In general, spline function with finite time difference approximation models need one previous time instance to accurately model the driver output voltage characteristics

$$f_1(k) = f_{1,s}(k) + p^* i'_{oh} + q^* i''_{oh} + \cdots$$
$$f_2(k) = f_{2,s}(k) + pp^* i'_{ol} + qq^* i''_{ol} + \cdots. \quad (8)$$

It can be seen from Fig. 3(c) that with the inclusion of one previous time instance of the driver output current (AGPV3V2), the modeled and simulated output current values match accurately. Once f_1 and f_2 are estimated for input HIGH and LOW, respectively, the relation between driver output current and voltage can be expressed as shown in

$$i_o(k) = w_1(k) f_1(k) + w_2(k) f_2(k) \quad (9)$$

where, w_1 and w_2 are the weighting functions that help f_1 and f_2 transition from one state to another. Since, two unknowns exist, weighting functions w_1 and w_2 can be found by linear inversion of (9) for two different loads as shown in (10). Fig. 4(a) and (b) show an example of the weighting functions w_1 and w_2 generated for IBM driver (AGPV3V2) model for a load of 50 Ω and 50 Ω + 1.5 V. Spline functions with finite time difference modeling technique is insensitive to the load connected at the driver output for a range of loads determined during the construction of the macromodel

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} f_{1a} & f_{2a} \\ f_{1b} & f_{2b} \end{bmatrix}^{-1} \begin{bmatrix} i_a \\ i_b \end{bmatrix}. \quad (10)$$

III. Modeling of Preemphasis Driver Circuits

Preemphasis (Precompensation) drivers are important in SI analysis of large digital systems. Preemphasis drivers are effective in driving signals though lossy transmission lines. These drivers boost the magnitude of high-frequency spectral components of signals, thus, ensuring that the signal reaches the receiver without affecting the logic even after attenuation. Preemphasis drivers are useful in reducing intersymbol interference (ISI). Fig. 5 shows a two-tap finite-impulse response (FIR) driver precompensation scheme, and its corresponding output waveform can be seen in Fig. 6. The amount of precompensation is shown by the parameter R in Fig. 5 which varies from 0.0 (no precompensation) to < 1.0 for higher levels of precompensation. Preemphasis drivers usually incorporate four voltage levels: HIGH, LOW, Strong HIGH, and Strong LOW. Preemphasis comes into effect only when the signal bit makes transition from one state to another. For example, in Fig. 6, when the signal

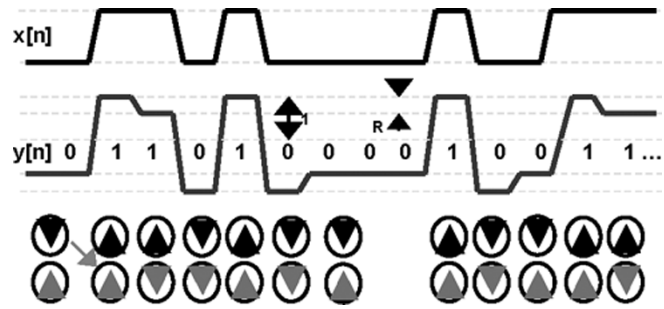


Fig. 6. Input and output voltage waveforms for a two-tap FIR pre-compensation driver.

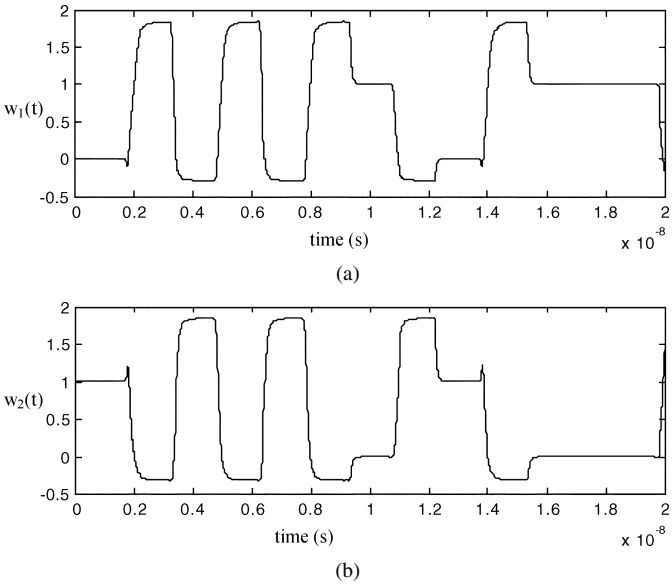


Fig. 7. (a) Weighting function w_1 for IBM preemphasis driver. (b) Weighting function w_2 for IBM preemphasis driver.

makes a transition from LOW to HIGH state, the signal level is boosted to a higher magnitude to reach strong HIGH. When the signal bit stays at the same logic level, pre-compensation stays the same and the signal stays in HIGH state. The same theory holds good for HIGH to LOW transition.

Modeling of preemphasis drivers is similar to modeling ordinary single-ended drivers. Submodels f_1 and f_2 in (8) that model the driver nonlinearity for input HIGH and LOW states, respectively, have to be carefully estimated. In case of a two-tap FIR precompensation driver, f_1 estimates the driver nonlinearity for HIGH state and f_2 estimates the nonlinearity for LOW state, but care should be taken in ensuring that the range of submodels f_1/f_2 should be from Strong LOW to Strong HIGH states. This ensures that submodels f_1/f_2 can capture the nonlinearity of the driver with preemphasis when the output voltage swings from Strong HIGH and *vice-versa*. The weighting functions that are calculated from linear inversion of (10) take into account the effect of preemphasis. Fig. 7(a) and (b) show the weighting functions w_1 and w_2 for IBM driver (BBPICMPTERM_A) for two different sets of loads as explained in the previous section. It can be seen from Fig. 7(a) and (b) that the weighting functions take into account the effect of preemphasis.

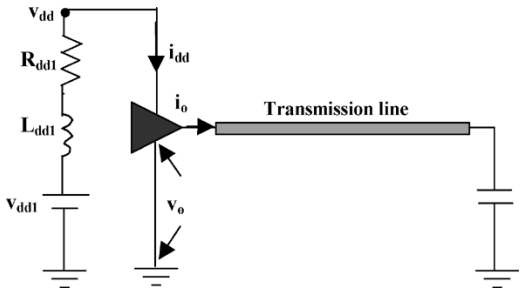


Fig. 8. Driver with a nonideal power supply v_{dd} is connected to a transmission line.

IV. Extension to Multiple Ports

The method discussed in the previous section can be extended to multiple ports. Fig. 8 shows a scenario where the power supply is nonideal. To incorporate the effect of the power supply node v_{dd}, a new relation should be drawn between driver power supply current (i_{dd}) and driver power supply voltage. However, the driver power supply current is not only a function of driver power supply voltage but also a function of driver output voltage (v_o), as shown in

$$i_{dd}(k) = w_{1,dd}(k) f_{1,dd}(v_o(k), v_{dd}(k)) + w_{2,dd}(k) f_{2,dd}(v_o(k), v_{dd}(k)) + w_{3,dd} \quad (11)$$

where, $f_{1,dd}$ and $f_{2,dd}$ are the power supply sub-models that relate the power supply current to power supply voltage for driver inputs HIGH and LOW respectively. In (11), weighting functions $w_{1,dd}$, $w_{2,dd}$, and $w_{3,dd}$ help sub-models $f_{1,dd}$ and $f_{2,dd}$ in transitioning from one state to another [11]. Sub-models $f_{1,dd}$ and $f_{2,dd}$ can be expressed as a combination of static sub-models ($f_{1,sd}$ and $f_{2,sd}$) and dynamic sub-models ($f_{1,dy}$ and $f_{2,dy}$), as shown in

$$f_{n,dd}(v_o(k), v_{dd}(k)) = f_{n,sd}(v_o(k), v_{dd}(k)) + f_{n,dy}(v_o(k), v_{dd}(k)); \text{ where, } n = 1, 2. \quad (12)$$

The static submodels $f_{1,sd}$ and $f_{2,sd}$ are calculated through a double *dc* sweep at driver output and at driver power supply. This relationship can be shown as

$$f_{n,sd}(v_o(k), v_{dd}(k)) = A_{n,r} {v_o}^r(k) {v_{dd}}^s(k) + A_{n,(r-1)} {v_o}^{(r-1)} {v_{dd}}^{(s-1)}(k) + \cdots + A_{n,0} \quad (13)$$

where A's are constants; $n = 1, 2$; $r >= 1$; $s >= 1$.

The dynamic submodels $f_{1,dy}$ and $f_{2,dy}$ can be expressed as

$$f_{n,dy}(v_o(k), v_{dd}(k)) = p_{n,d} \Delta i'_{n,o} + q_{n,d} \Delta i'_{n,dd} + pp_{n,d} \Delta i''_{n,o} + qq_{n,d} \Delta i''_{n,dd} \cdots, \qquad n = 1, 2. \quad (14)$$

In (14), dynamic characteristic submodel ($f_{n,dy}$) constants $p_{n,d}$, $q_{n,d}$, $pp_{n,d}$ and $qq_{n,d}$ are estimated by connecting PWL voltage sources both at driver output and at driver power supply and measuring the error between static power supply current

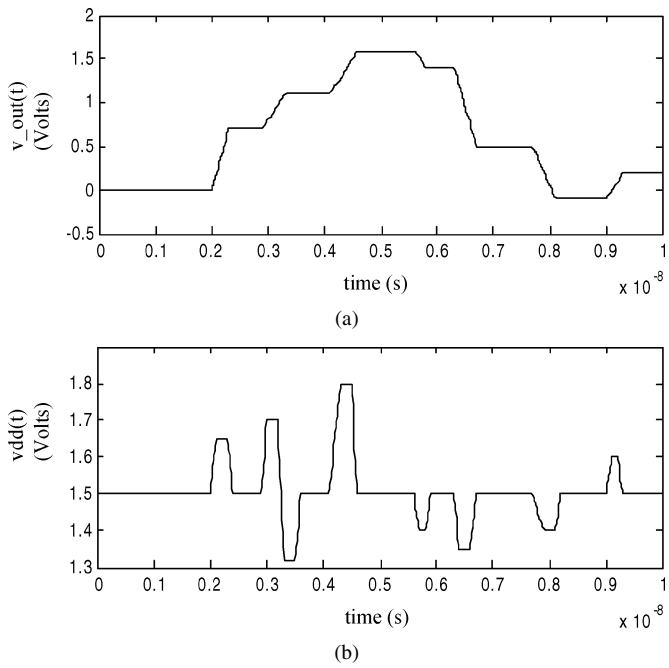


Fig. 9. (a) PWL voltage sources connected at the driver output to calculate the dynamic characteristics of spline function with finite time difference approximation. (b) PWL voltage sources connected at the driver power supply node to calculate the dynamic characteristics of spline function with finite time difference approximation.

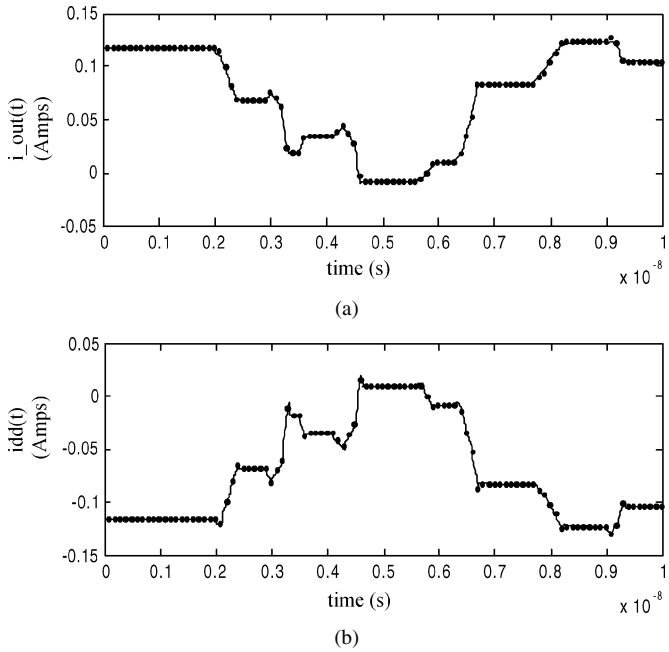


Fig. 10.(a) Driver output current current from transistor level driver (straight line) and spline model with finite time difference approximation (dotted line) when the driver input is HIGH. (b) Driver power supply current from transistor level driver (straight line) and spline model with finite time difference approximation (dotted line) when the driver input is HIGH.

and transistor level driver power supply current. Fig. 9(a) and (b) show PWL voltage sources connected at driver output and power supply, respectively, for IBM driver (HSTL_A) when the driver input is held HIGH. Fig. 10(a) and (b) shows the resultant IBM driver output current and the power supply current respectively. It can be also seen from Fig. 10(a) and (b) that the

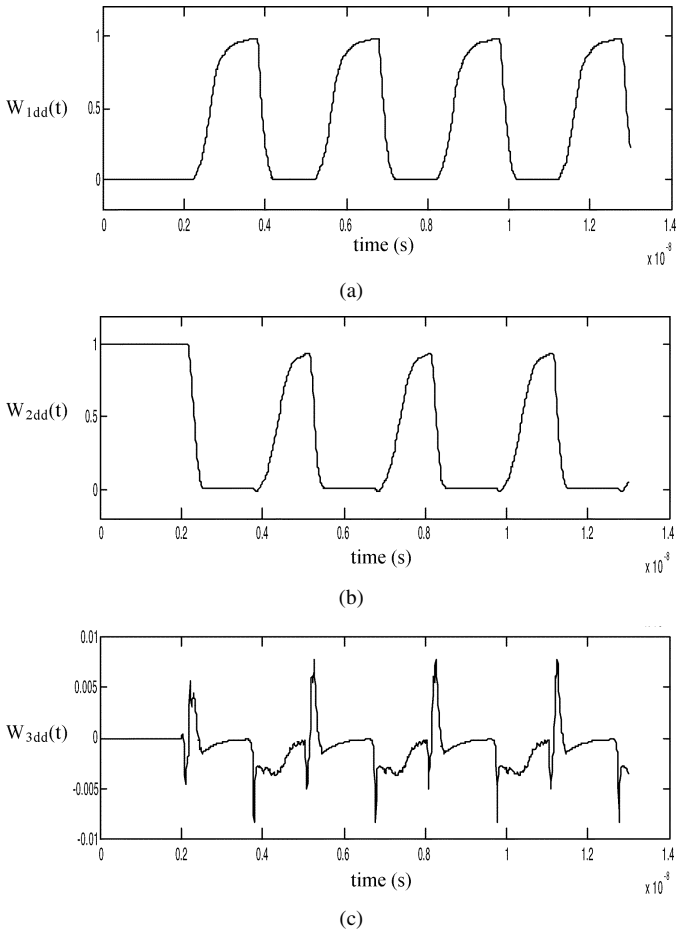


Fig. 11. (a) Weighting functions w_{1dd} that help submodels f_{1dd} and f_{2dd} transition from one state to another. (b) Weighting functions w_{2dd} that help submodels f_{1dd} and f_{2dd} transition from one state to another. (c) Weighting functions w_{3dd} that help submodels f_{1dd} and f_{2dd} transition from one state to another.

spline model with finite time difference approximation models the driver output and power supply current accurately. In (13) and (14), A_n, $p_{n,d}$, $q_{n,d}$, $pp_{n,d}$ and $qq_{n,d}$ are all constants and depend on the driver being modeled. Terminating the driver with two different loads results in two equations of (11) and since, three unknown weighting functions exist, one method for solving the problem is by assuming $w_{1,dd} = (1 - w_{2,dd})$. Thus, weighting functions $w_{1,dd}$, $w_{2,dd}$, and $w_{3,dd}$ can be calculated once $f_{1,dd}$ and $f_{2,dd}$ are estimated for two different loads [11]. Fig. 11(a)–(c) show the weighting functions $w_{1,dd}$, $w_{2,dd}$, and $w_{3,dd}$, respectively, for the IBM driver.

Similarly, driver output current is not only a function of driver output voltage but also a function of driver supply voltage. The procedure for estimating driver output current is similar to estimating driver power supply current. Equations (15)–(18) provide the relation between driver output current in terms of driver output voltage and driver power supply voltage.

$$\begin{aligned} i_o(k) &= w_1(k) f_1(v_o(k), v_{dd}(k)) \\ &\quad + w_2(k) f_2(v_o(k), v_{dd}(k)) \end{aligned} \tag{15}$$

$$\begin{aligned} f_n(v_o(k), v_{dd}(k)) &= f_{n,s}(v_o(k), v_{dd}(k)) \\ &\quad + f_{n,d}(v_o(k), v_{dd}(k)) \end{aligned} \tag{16}$$

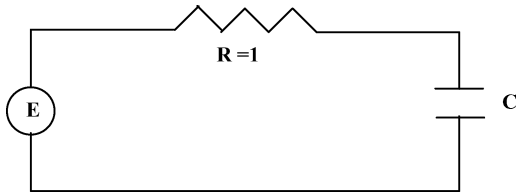


Fig. 12. Circuit equivalent representation of dynamic characteristics.

$$f_{n,s}(v_o(k), v_{dd}(k)) = A_{n,t} {v_o}^t(k) {v_{dd}}^u(k) + A_{n,(t-1)} {v_o}^{(t-1)}(k) \cdot {v_{dd}}^{(u-1)}(k) + \cdots + A_{n,0} \quad (17)$$

where $n = 1, 2;\ t >= 1$, and $u >= 1$

$$f_{n,d}(v_o(k), v_{dd}(k)) = p_n \Delta i'_{n,o} + q_n \Delta i'_{n,dd} + pp_n \Delta i''_{n,o} + qq_n \Delta i''_{n,dd} \ldots, \qquad n = 1, 2 \quad (18)$$

where p_n, q_n, pp_n, and qq_n are constants.

V. SPICE Circuit Implementation

A SPICE equivalent circuit for the spline function macromodel with finite time difference approximation can be generated using voltage-dependent current sources and voltage-dependent voltage sources. The weighting functions w_1, w_2, $w_{1,dd}$, $w_{2,dd}$, and $w_{3,dd}$ that are calculated by linear inversion of (11) and (15) can be represented using a PWL voltage source. A voltage-dependent current source has been used to represent submodels f_1, f_2, $f_{1,d}$, , and $f_{2,d}$. The dynamic behavior of the driver has been captured using state equations. Assuming

$$E(k) = f_1(k) \quad (19)$$

$$\text{Since} \quad f_1(k) - f_1(k-1) = f_1(t) - f_1(t - \Delta t) = \Delta i_{oh} \quad (20)$$

$$\frac{E(t) - E(t - \Delta t)}{\Delta t} = \frac{\Delta i_{oh}}{\Delta t}. \quad (21)$$

Fig. 12 shows the equivalent circuit representation of (20) and (21) for capturing the dynamic characteristics (previous time instances). The value of capacitor "C" is based on the sampling time step. Depending on the driver, "C" can vary from 5 to 100 pF. Similarly, the above technique can be used to capture the previous time instances for f_2, $f_{1,d}$, and $f_{2,d}$. The SPICE netlist can also be generated for capturing other previous time instances of f_1, f_2, $f_{1,d}$, and $f_{2,d}$ in a similar manner.

VI. RBF Modeling Method

As a comparison, in RBF modeling method, a nonlinear relation is drawn between the driver output current and the driver output voltage using radial basis functions in a piece-wise (P-W) equation as shown in (22) [10]

$$i_o(k) = w_1(k) f_1(\Theta_1, x(k)) + w_2(k) f_2(\Theta_2, x(k)) \quad (22)$$

$$f_n(\Theta_n, x(k)) = \sum_{j=1}^{M} \theta_{nj} \phi(|x - cnj|, \beta), \qquad n = 1, 2 \quad (23)$$

where i_o is the driver output current, and f_1 and f_2 are the RBF submodels that relate the driver output current to the driver output voltage for driver input HIGH and LOW, respectively. The functions f_1 and f_2 are nonlinear dynamic parametric models. They consist of sums of Gaussian basis functions that depend on the past r (dynamic order of the model) samples of the output current and present and past r samples of voltage v_o. Submodels f_1 and f_2 are parameterized using effective estimation algorithms [8], [9]. A more in-depth explanation of the modeling technique is available in [10]. The time varying weighting functions w_1 and w_2 help in transitioning submodels f_1 and f_2 from one state to another. The weighting functions w_1 and w_2 are computed by linear inversion of (22), for two different loads. Submodels f_1 and f_2 can be expressed as shown in (23), where M is the number of basis functions needed for f_1 or f_2 to accurately model the digital driver. Φ is the asymptotically increasing or decreasing basis function and θ_j is the weight of the basis function. The centers of the basis functions are defined by c_j and the width or the spread parameter is defined by β [10]. The *regressor vector* x collects the past r samples of the driver output voltage and the driver output current along with the present sample of the driver output voltage as shown in (24).

$$x^T(k) = \left\{ \begin{matrix} i_o(k-1), i_o(k-2), \ldots, i_o(k-r), \\ v_o(k), v_o(k-1), \ldots\ldots\ldots, v_o(k-r) \end{matrix} \right\}. \quad (24)$$

VII. Test Results

In this section, the accuracy and the simulation speed between the method proposed and transistor level driver models has been studied for different test cases.

A. Two Port Macromodels

Example 1: A test vehicle was designed to verify the accuracy of spline model with finite time difference approximation method. The proposed method was also compared with the RBF based model. The driver used was an IBM encrypted SPICE transistor level model (BagpV3V2) with a 1.5-V power supply and consumed 800 KB of memory. The IBM driver was connected to a 75-Ω ideal transmission line with a delay of 0.2 ns. The transmission line was terminated with a 1-pF load. A pulse with 0.2 ns rise time and a period of 2 ns was used. The near and far end voltage waveforms on the transmission line were measured as shown in Fig. 13. RBF model used ten basis functions to model f_1 and eight to model f_2. The dynamic order r used for RBF model was 1. A third-order spline function with one previous time instance was needed for submodels f_1 and f_2 to accurately model the driver. The values of p and pp were $5*$ sampling time which was 20 ps. IBM transistor level driver model took 15 min 34 s for simulation, RBF model took 35 s, and spline model with finite time difference took 14 s for simulation. All simulations were carried out on a SUN ULTRA-10 workstation.

The amount of simulation time consumed by IBM transistor level driver model (BagpV3V2), RBF model, and spline model with finite time difference approximation were compared when the rise time of the driver was reduced. The rise time of the driver was decreased from 1 ns to 0.2 ns, and the simulation time consumed for all the three models are plotted as shown in Fig. 14. It can be seen that both RBF and spline models are faster

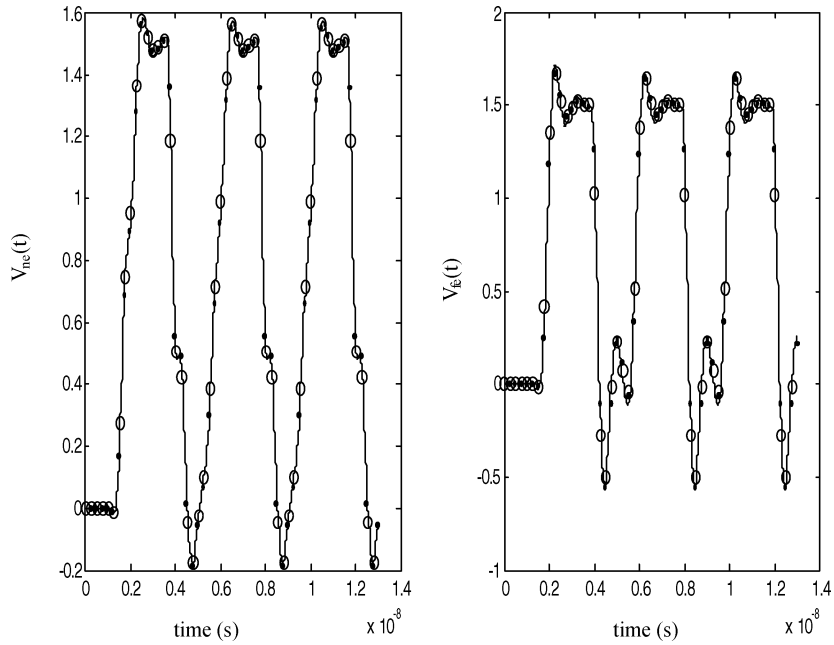


Fig. 13. Near end $V_{ne}(t)$ and far end $V_{fe}(t)$ voltage waveforms for IBM driver (straight line), Spline function with finite time difference (dotted line) and RBF model (0).

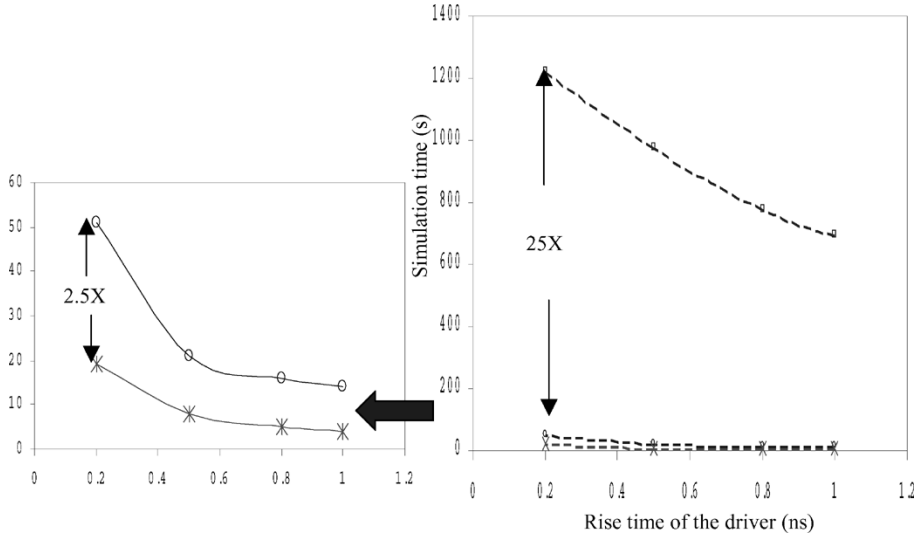


Fig. 14. Simulation time consumed as the driver rise time decreases by transistor level IBM BagpV3V2 driver (□), RBF model (○), and spline model with finite time difference approximation (*).

than actual transistor level driver model. It can also be noticed that spline model is two to three times faster than RBF model.

Example 2: In this test case, crosstalk between transmission lines was calculated using both IBM transistor level driver model (HSTL_A) and spline model with finite time difference approximation. The accuracy of the spline model in capturing sensitive effects like crosstalk was tested using this example. The test setup consisted of 11 identical drivers, each of them connected to a 25-Ω w-element transmission line which was terminated using a 1-pF capacitance. All the transmission line models were generated using H-SPICE field solver. The transmission lines were microstrip structures with distance between adjacent lines being 0.6 mm. All the drivers except driver 4 and driver 8 were switching simultaneously. The drivers have an input rise time of 0.1 ns at 400 MHz. Driver 4 was set idle at zero level and driver 8 was set idle at 1.5 V. The voltage waveforms at the near end of the transmission lines connected to drivers 4 and 8 are shown in Fig. 15. It can be seen that spline model with finite time difference accurately captures the crosstalk between transmission lines accurately. A third order spline function with one previous time instance was needed for submodels f_1 and f_2 to accurately model the driver. The value of p and pp was $4*$ sampling time which was 10 ps.

Example 3: A test case was designed to test the accuracy of spline function modeling technique in modeling preemphasis driver circuits. Spline function modeling technique was used to model an IBM driver (BBPICMPTERM_A) that had a power supply voltage of 1.2 V and a rise time of 50 ps. The driver

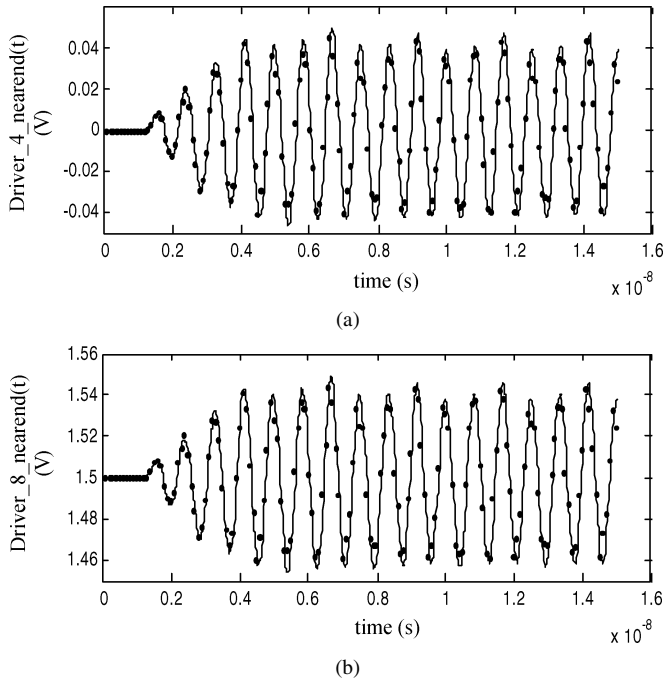


Fig. 15. (a) Crosstalk effect at the near end of transmission lines for driver 4 from IBM transistor level driver model (straight line) and spline function with finite time difference approximation model (dotted line). (b) Crosstalk effect at the near end of transmission lines for driver 8 from IBM transistor level driver model (straight line) and spline function with finite time difference approximation model (dotted line).

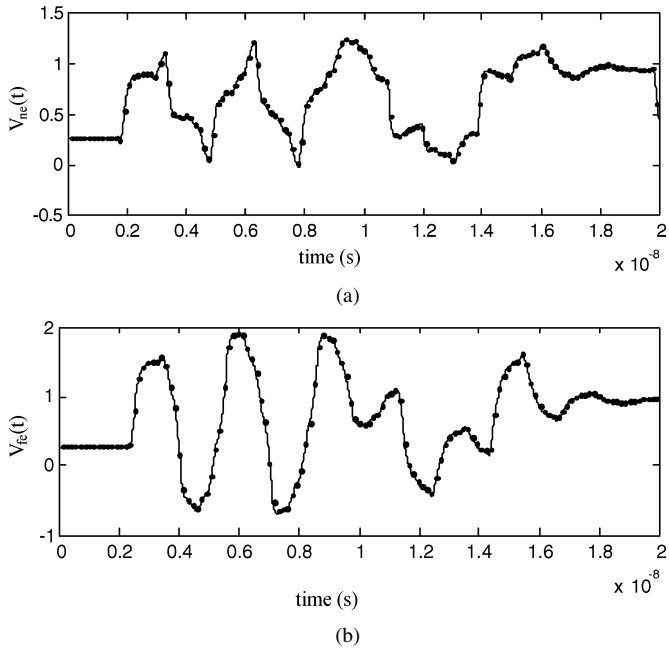


Fig. 16. (a) Voltage waveform at the near end (V_ne(t)) of the transmission line from IBM transistor-level driver (straight line) and spline function with finite time difference approximation model (dotted line). (b) Voltage waveform at the far end (V_fe(t)) of the transmission line from IBM transistor-level driver (straight line) and spline function with finite time difference approximation model (dotted line).

was connected to a 50-Ω ideal transmission line with a delay of 0.5 ns which in turn was terminated with a 2-pF capacitance. A random bit pattern (010101100111...) was given to the driver, and voltage waveforms at the near end and far end of the transmission lines were measured. It can be seen from Fig. 16(a) and (b) that spline function with finite time difference macromodel accurately captures the nonlinearity of the driver. A third-order cubic polynomial was used in submodels f_1 and f_2. The value of p and pp were $10*$ sampling time and $9*$ sampling time, respectively. A sampling time of 10 ps was used for this test case. Spline function with finite time difference macromodel was 47 times faster than the IBM transistor-level driver model.

B. Four-Port Macromodels

Example 1: In this test case, an IBM driver HSTL_B was connected to a plane pair which was modeled using the cavity resonator method [12]. In cavity resonator method, the impedance between two ports on the plane can be expressed as

$$Z_{ij}(w) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{N_{mni}N_{mnj}}{jwC_{mn} + \frac{1}{jwL_{mn}} + G_{mn}}. \tag{25}$$

The equivalent circuit for (25) can be implemented by using parallel resonant circuits and ideal transformers [12]. The plane pair was 10 $\times$ 6 cm in length and width. It had five ports on each plane, V_{dd} and *Gnd* as shown in Fig. 17. Ten drivers were connected at port 1 and all the drivers were connected to 25-Ω ideal transmission lines which in turn were terminated using 1-pF capacitance. All transmission lines were terminated at port 3. The power supply (v_{dd}) was at port 4. Two additional ports were used for probing. All 10 drivers were identical. The resulting simultaneous switching noise (SSN) was calculated using both actual transistor-level driver model and spline function with finite time difference approximation model. Fig. 18(a) shows the near end and far end voltage waveforms on transmission line 1. It can be clearly seen that the modeled and the transistor-level model waveforms match accurately. A third-order spline function with two previous time instances was used to model submodels f_1 and f_{1dd}. Fig. 18(b) shows the SSN at ports 1, 3, and 5, respectively both for modeled and actual simulated cases, and again it can be clearly seen that the noise is estimated accurately using the spline function with finite time difference approximation method. Spline function model was eight times faster than the IBM transistor level driver model.

Example 2: A comparison study was done between simulation time taken by the IBM transistor-level driver circuit (HSTL_A) and spline function with finite difference approximation model when the number of drivers switching was increased from 1 to 32. Fig. 19(a) shows the plot for simulation time consumed versus number of drivers switching, and Fig. 19(b) shows the percentage peak noise error when the number of drivers switching is increased from 2 to 32.

It can be seen that when 32 drivers are switching, the spline model is 25–30 times faster than the transistor-level driver circuit. All the simulations were carried out on a SUN ULTRA 10 machine for 30-ns time period. The percentage peak noise error between the transistor-level model and spline function model was calculated when the number of the drivers switching was increased from 1 to 32. It can be clearly seen that when 32 drivers are switching, the percentage peak noise error is less than 4.5%. From these results it can be seen that spline model with finite time difference approximation method can accurately model SSN and reduce the computation time.

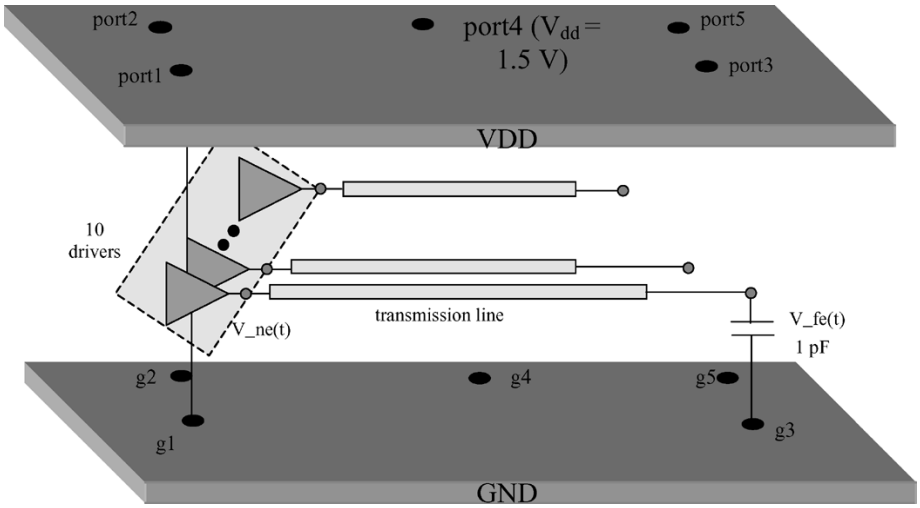


Fig. 17. Plane pair model generated using cavity resonator method. Both planes have six ports each.

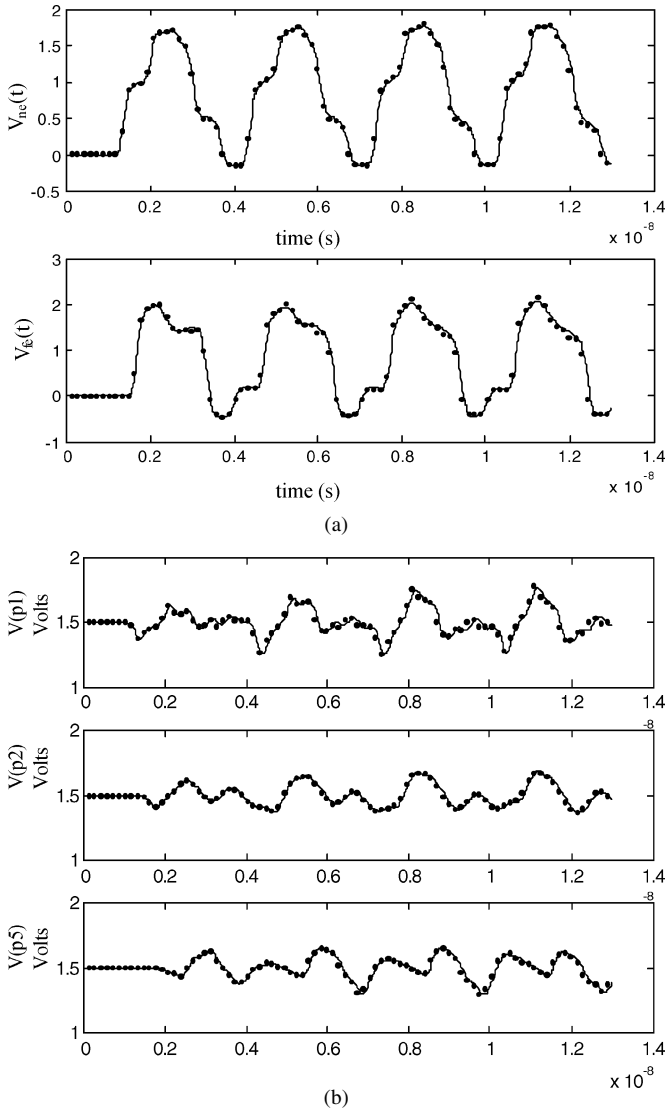


Fig. 18. (a) Near end (V_ne(t)) and far end (V_fe(t)) voltage waveforms on transmission line 1 for IBM transistor-level driver model (straight line) and spline function with finite time difference approximation model (dotted line). (b) SSN at ports p1, p2, and p5 when ten identical drivers were switching together. SSN from actual transistor-level driver model (straight line) and spline function with finite time difference approximation model (dotted line).

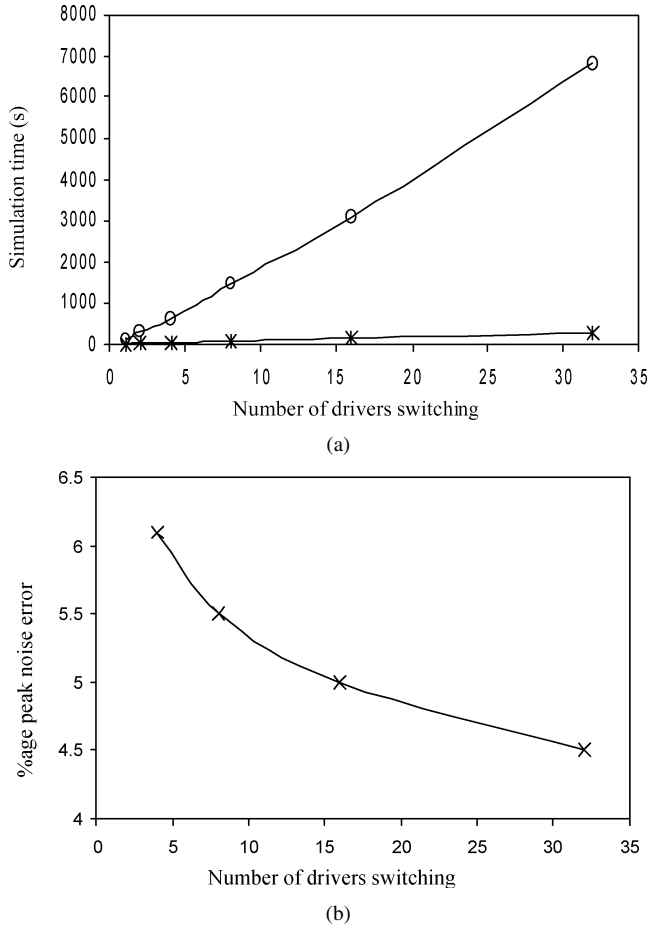


Fig. 19. (a) Simulation time versus number of drivers switching for transistor level driver (0) and spline model with finite time difference approximation (*). (b) Percentage peak noise error versus number of drivers switching for spline model with finite time.

VIII. Limitations

Spline function with finite time difference approximation has some limitations. When the transistor-level driver circuit models are highly complex, the proposed method cannot capture the high nonlinearity present in the driver. Hence, spline function

with finite time difference models are more suitable for moderately nonlinear drivers. One solution for modeling these highly nonlinear driver circuits is the use of RNN. RNN models are artificial neural network models that are used to model highly nonlinear patterns with memory or feedback. RNN functions can model the nonlinearity of these complex drivers. Submodels f_1 and f_2 in (8) can now be expressed using RNN functions as shown in (26) as follows:

$$f_n = \sum_{k=1}^{M} b_{kj} g\left(\sum_{j=1}^{N} a_{ji} x_i + a_{oj}\right) + b_{ok}\,, \qquad n = 1, 2 \tag{26}$$

where

$$g(x) = \frac{(e^x + e^{-x})}{(e^x - e^{-x})}. \tag{27}$$

In (26) and (27), b and a are weights associated with the neural network, N represents number of hidden neurons, M represents number of outputs, and x is the regressor vector as shown in (24) [13]. In RNN training, all weights in all of the layers are adjusted until the required accuracy is achieved; but in RBF training, however, the RBF centers are picked one at a time from the data set minimizing the least mean square error (LMSE) until the required accuracy is achieved. RBF functions use localized Gaussian functions where as RNN uses globalized hyperbolic tangential functions. RNN use back propagation through time (BPTT) algorithm to estimate the weights [15]. This algorithm takes into account the feedback effect of the output in training the neural network, which makes RNN functions more robust in modeling highly nonlinear signatures. Depending on the nonlinearity of the driver to be modeled, neural networks can be extended to include multiple hidden layers and increase hidden neurons.

Example 1: An IBM (SDRAM) transistor-level driver model was connected to a 50-Ω ideal transmission line and terminated with a 2-pF capacitance. The driver was given an input pulse with 0.2-ns rise time at 200 MHz. The voltage waveforms at the near and far end of the transmission lines were measured from RBF and spline function with finite time approximation models. A fourth-order polynomial was used in spline function with finite time difference model. Submodels f_1 and f_2 in RBF models needed eight and nine basis functions, respectively. It can be seen from Fig. 20(a) and (b) that both models failed to accurately capture the nonlinearity of the driver. For RNN model, submodels f_1 and f_2 needed theee hyperbolic tangent functions each with two previous time instances of driver output current and output voltage. It was found that RNN model gives high accuracy for the same test case as shown in Fig. 21(a) and (b).

Example 2: In this test case, four IBM drivers (DDR2) were connected to a small plane pair modeled using cavity resonator method. IBM DDR2 driver has a power supply voltage of 2.5 V and it is usually driven at 250 MHz with a rise time of 1.25 ns. The plane pair had dimensions of 6 $\times$ 4 cm with four ports on V_{dd} plane and four ports on *Gnd* plane. All the drivers were identical, driving ideal 50-Ω transmission lines and were connected

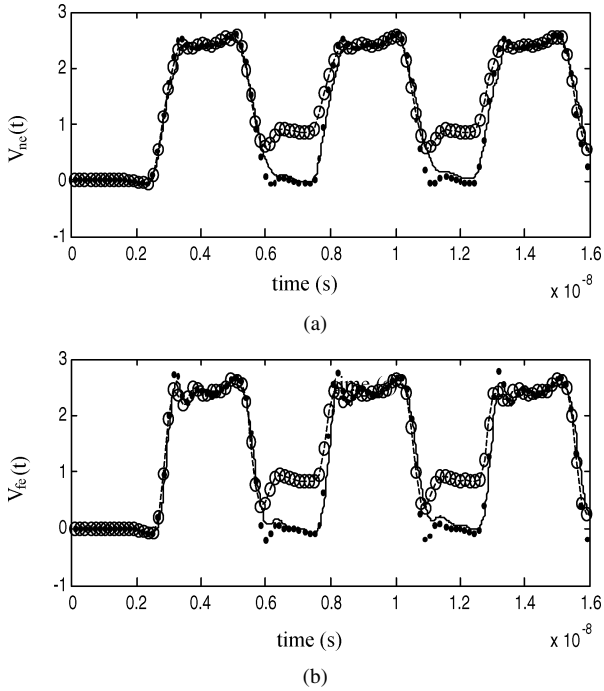


Fig. 20. (a) Near end (V_ne(t)) voltage waveform on transmission line for IBM transistor level driver model (straight line), RBF model (circles), and spline model with finite time difference approximation (dotted line). (b) Far end (V_fe(t)) voltage waveform on transmission line for IBM transistor-level driver model (straight line), RBF model (circles), and spline model with finite time difference approximation (dotted line).

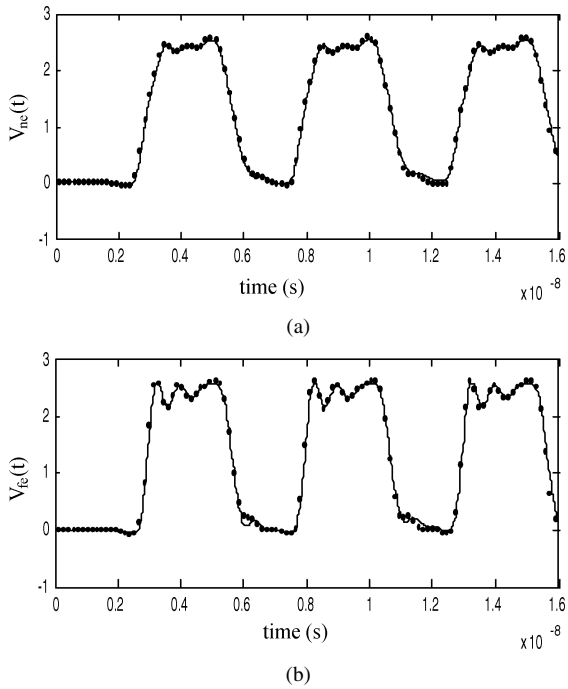


Fig. 21. (a) Near end (V_ne(t)) voltage waveform on transmission line for IBM transistor-level driver model (straight line) and RNN model (dotted line). (b) Far end (V_fe(t)) voltage waveform on transmission line for IBM transistor-level driver model (straight line) and RNN model (dotted line).

at port 1. The power supply node was at port 3. All the transmission lines were terminated at port 2 using a 2-pF capacitance and port 4 was used for probing. Fig. 22 shows the plane pair used to

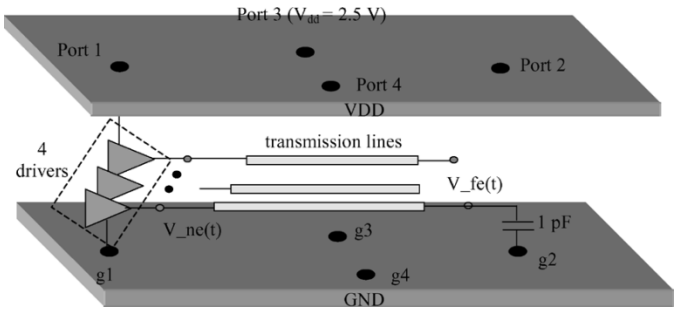


Fig. 22. Plane pair model generated using cavity resonator method. Both planes have four ports each.

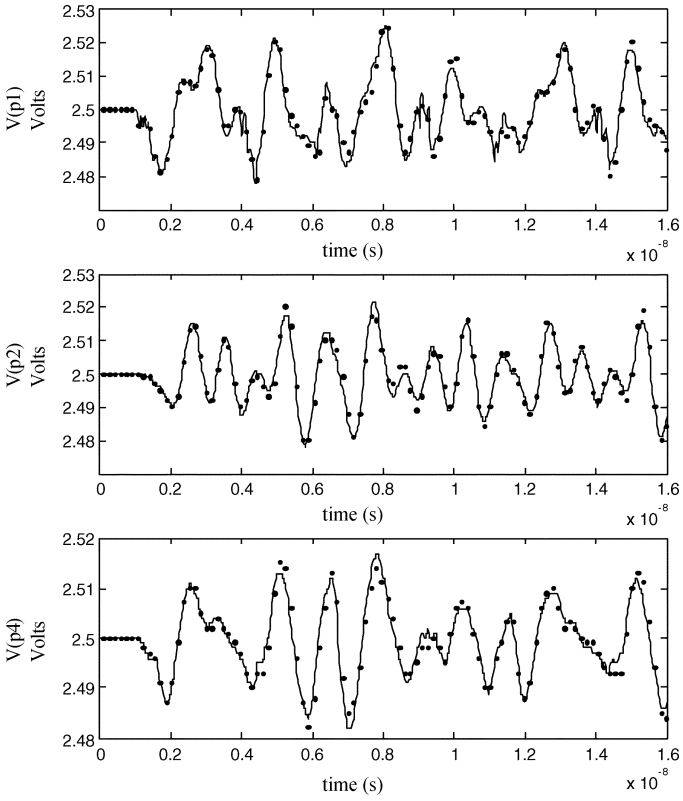


Fig. 23. SSN at ports p1, p2 and p4 when four identical drivers were switching together. SSN from actual transistor-level driver model (straight line) and RNN model (dotted line).

model the power supply noise when four drivers are simultaneously switching. The IBM driver was modeled using RNN. The regressor vector x consists of present samples of power supply voltage and driver output voltage and past samples of driver output current, power supply current, power supply voltage, and output voltage. The RNN model for subfunctions $f_{1d,2d}$ require one hidden layer with two hidden neurons. BPTT training algorithm was used to estimate the weights of the RNN model [15]. It can be seen from Fig. 23 that RNN model captures SSN accurately when all four drivers are switching simultaneously. RNN model is 6–7 times faster than the transistor-level model when the time domain simulations were carried out for three cycles. The computational time speed up between the RNN model and actual transistor-level model increases with the increase in complexity of the circuit.

IX. Conclusion

A modeling method using spline function with finite time difference approximation has been proposed in this paper. Although RBF modeling technique is accurate, it becomes complicated to parameterize RBF models when the method is extended to multiple ports resulting in SPICE convergence problems. It has been shown in this paper that spline function with finite time difference modeling technique is simple and accurate for modeling the driver circuits discussed. It can be extended to multiple ports and can accurately capture sensitive effects like SSN and crosstalk accurately. Spline function with finite time difference approximation models are faster than transistor-level driver models, consume less memory, and are less susceptible to numerical convergence problems with SPICE. For highly nonlinear drivers where spline models with finite time approximation fail to capture the nonlinearity, RNN models can be used as a good replacement to spline models.

References

[1] I/O Buffer Information Specification (IBIS) Ver. 3.2 (1999, Sep.). [Online]. Available: http://www.eigroup.org/ibis/- ibis.htm

[2] "Stat-HSPICE manual 2001.2," in *Performing Behavioral Modeling*, ch. 26.

[3] A. Varma, A. Glaser, S. Lipa, M. Steer, and P. Franzon, "The development of a macro-modeling tool to develop IBIS models," in *Proc. 12th Topical Meeting Electrical Performance of Electronic Packaging (EPEP)*, Princeton, NJ, Oct. 2003, pp. 277–280.

[4] I. S. Stievano and I. A. Maio, "Behavioral models of digital IC ports from measured transient waveforms," in *Proc. 9th IEEE Topical Meeting Electrical Performance Electronic Packaging (EPEP)*, Scottsdale, AZ, Oct. 23–25, 2000, pp. 211–214.

[5] I. S. Stievano, Z. Chen, D. Becker, F. G. Canavero, G. Katopis, and I. A. Maio, "Behavioral modeling of digital IC input and output ports," in *Proc. 10th Topical Meeting Electrical Performance Electronic Packaging (EPEP)*, Cambridge, MA, Oct. 29–31, 2001, pp. 331–334.

[6] I. S. Stievano, F. G. Canavero, Z. Chen, G. Katopis, and I. A. Maio, "Parametric macromodels of drivers for SSN simulations," in *Proc. IEEE Int. Symp. EMC*, Boston, MA, Aug. 18–22, 2003, pp. 616–621.

[7] B. Mutnury, M. Swaminathan, and J. Libous, "Macro-modeling of nonlinear I/O drivers using spline functions and finite time difference approximation," in *Proc. 12th Tropical Meeting Electrical Performance Electronic Packaging (EPEP)*, Princeton, NJ, Oct. 2003, pp. 273–276.

[8] K. Judd and A. Mees, "On selecting models for nonlinear time series," *Physica D*, vol. 82, pp. 426–444, 1995.

[9] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function network," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.

[10] I. S. Stievano, I. A. Maio, and F. G. Canavero, "Parametric macromodels of digital I/O ports," *IEEE Trans. Adv. Packag.*, vol. 25, no. 2, pp. 255–264, May 2002.

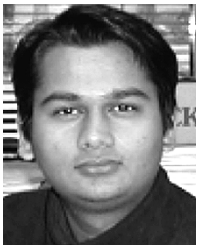
[11] B. Mutnury, M. Swaminathan, and J. Libous, "Black-box modeling of nonlinear I/O drivers," in *Proc. 20th Annu. Review Progress in Applied Computational Electromagnetics (ACES)*, Syracuse, NY, Apr. 19–23, 2004.

[12] S. Chun, M. Swaminathan, L. D. Smith, J. Srinivasan, Z. Jin, and M. K. Iyer, "Modeling of simultaneous switching noise in high speed systems," *IEEE Trans. Adv. Packag.*, vol. 24, no. 2, pp. 132–142, May 2001.

[13] S. Haykin, *Neural Networks – A Comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[14] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Boston, MA: Artech House, c2000.

[15] Y. Fang, M. C. E. Yagoub, F. Wang, and Q. J. Zhang, "A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 12, pp. 2335–2344, Dec. 2000.

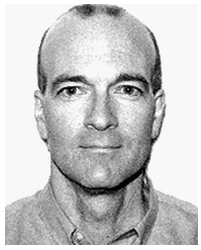


Bhyrav Mutnury (S'00) received the B.Sc. degree from JNTU College of Engineering, Hyderabad, India, in 2000 and the M.S degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in 2002, where he is currently pursuing the doctoral degree in electrical engineering under the supervision of Dr. Madhavan Swaminathan. His current study is on nonlinear driver I/O macromodeling.

Madhavan Swaminathan (A'91–M'95–SM'98) received the M.S. and Ph.D. degrees in electrical engineering from Syracuse University, Syracuse, NY.

He is currently a Professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, and the Deputy Director of the Packaging Research Center, Georgia Tech. He is the cofounder of Jacket Micro Devices, Atlanta, a company specializing in integrated devices and modules for wireless applications where he serves as the Chief Scientist. Prior to joining Georgia Tech, he was with the Advanced Packaging Laboratory at IBM, Fishkill, NY, working on packaging for super computers. He has over 150 publications in refereed journals and conferences, has co-authored three book chapters, has nine issued patents and has six patents pending. His research interests are in digital, RF, optoelectronics and mixed-signal packaging with emphasis on design, modeling, characterization, and test.

Dr. Swaminathan reached the second invention plateau while at IBM. He served as the Co-Chair for the 1998 and 1999 IEEE Topical Meeting on Electrical Performance of Electronic Packaging (EPEP), seerved as the Technical and General Chair for the IMAPS Next Generation IC and Package Design Workshop, serves as the Chair of TC-12, the Technical Committee on Electrical Design, Modeling, and Simulation within the IEEE CPMT society, and was the Co-Chair for the 2001 IEEE Future Directions in IC and Package Design Workshop. He is the co-founder of the IMAPS Next Generation IC and Package Design Workshop and the IEEE Future Directions in IC and Package Design Workshop. He also serves on the technical program committees of EPEP, Signal Propagation on Interconnects workshop, Solid State Devices and Materials Conference (SSDM), Electronic Components and Technology Conference (ECTC), and Interpack. He has been a Guest Editor for the IEEE TRANSACTIONS ON ADVANCED PACKAGING and IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES. He was the Associate Editor of the IEEE TRANSACTIONS ON COMPONENTS AND PACKAGING TECHNOLOGIES. He is the recipient of the 2002 Outstanding Graduate Research Advisor Award from the School of Electrical and Computer Engineering, Georgia Tech, and the 2003 Outstanding Faculty Leadership Award for the mentoring of graduate research assistants from Georgia Tech. He is also the recipient of the 2003 Presidential Special Recognition Award from IEEE Components, Packaging, and Manufacturing Technology Society for his leadership of TC-12 and the 2004 IBM Faculty Award. He has also served as the coauthor and advisor for a number of outstanding student paper awards at EPEP '00, EPEP '02, EPEP '03, EPEP '04, ECTC '98, and the 1997 IMAPS Education Award. He is the recipient of the Shri. Mukhopadyay Best Paper Award at the International Conference on Electromagnetic Interference and Compatibility (INCEMIC), Chennai, India, in 2003.



James P. Libous (SM'02) received the B.S. degree in electrical engineering from Union College, Schenectady, NY, and the M.S. degree in electrical engineering from Syracuse University, Syracuse, NY, where he is currently pursuing the Ph.D. degree in electrical engineering.

He is currently is a Senior Technical Staff Member in IBM's System and Technology Group, Endicott, NY. Since 1981, he has worked in several areas within IBM, including logic and memory integrated circuit development, system signal and power integrity and noise containment design, CMOS ASIC technology development, and complex project and engineering team leadership. He was responsible for the development and application of advanced CMOS VLSI circuit and package technologies for System/390 midrange processors. In 1992, he joined the Advanced CMOS ASIC Technology Development Group, Microelectronics Division. He was a key member of the team responsible for the development of the first one-million circuit CMOS ASIC logic family. He was responsible for the development of several advanced nanometer CMOS ASIC products, leading cross-divisional teams of development engineers and serving as a consultant to senior management. He was also responsible for identifying strategic package technology requirements to complement future nanometer CMOS ASIC and microprocessor products. He currently is a Consultant to senior management and provides technical leadership, guidance, and consulting in advanced nanometer CMOS chip and package electrical design. He holds U.S. Patents in the area of CMOS ASIC design and delta-I noise avoidance and has published several technical papers in refereed journals and conference proceedings. His research interests include signal and power distribution electrical design and modeling of advanced nanometer CMOS circuits, chips, and packages, including design for noise avoidance, and switching noise modeling, simulation, and laboratory characterization.

Mr. Libous serves on the Semiconductor Research Corporation Interconnect and Packaging Sciences and Student Relations Technical Advisory Boards. He serves as a Liaison/Mentor for SRC-funded research at Georgia Tech and IBM-funded research at Union College, Schenectady, NY, where he established undergraduate SRC research. He is the Recipient of the 2002 Semiconductor Research Corporation Mahboob Khan Outstanding Mentor Award, a lifetime award which recognizes individuals who have had a significant impact on the technical contributions of SRC research, have been instrumental in transferring the research results to industry, and are deeply committed to the education of the graduate students. He contributes to the SIA and NEMI technology roadmaps and has participated in Georgia Tech's Packaging Research Center, Cornell University's Packaging Alliance, and the Interconnect Focus Center Workshop at Albany NanoTech. He was the Recipient of the Shri Mukhopadhyay Best Paper Award at the International Conference on Electromagnetic Interference and Compatibility in 2003. He serves on the IBM Invention Review Board and received the IBM Division Award, several formal and informal awards, and Invention Achievement Awards. He was the Recipient of IBM Valuable Patent Awards in 2002 and 2003 for having patents in the top 5% of the IBM Microelectronics patent portfolio. He was an invited speaker at the IMAPS Workshop on Next Generation IC and Package Technology and at the Integrated Electronics Engineering Center (IEEC) at the Watson School of Engineering, Binghamton University. He also serves on the IBM System and Technology Group Electronic Package Technology Technical Community Council and has served on task forces conducted by the IBM Academy of Technology. He was Co-Chair for the InterPack Electrical Design, Simulation, and Test track. He serves on the TC-EDMS, the technical committee on Electrical Design, Modeling, and Simulation, and TC-NANO, Nano Packaging, both within the IEEE CPMT Society. He has been a session chair and short course instructor for the IEEE Topical Meeting on Electrical Performance of Electronic Packaging. He is a Guest Editor and Reviewer for the IEEE TRANSACTIONS ON ADVANCED PACKAGING. He also serves on the International Advisory Committee for the IEEE Electrical Design for Advanced Packaging and Systems (EDAPS).