

Compact Current Source Models for Timing Analysis under Temperature and Body Bias Variations

Saket Gupta, and Sachin S. Sapatnekar, *Fellow, IEEE*,

Abstract—State-of-the-art timing tools are built around the use of current source models (CSMs), which have proven to be fast and accurate in enabling the analysis of large circuits. As circuits become increasingly exposed to process and temperature variations, there is a strong need to augment these models to account for thermal effects and for the impact of adaptive body biasing, a compensatory technique that is used to overcome on-chip variations. However, a straightforward extension of CSMs to incorporate timing analysis at multiple body biases and temperatures results in unreasonably large characterization tables for each cell. We propose a new approach to compactly capture body bias and temperature effects within a mainstream CSM framework. Our approach features a table reduction method for compaction of tables and a fast and novel waveform sensitivity method for timing evaluation under any body bias and temperature condition. On a 45nm technology, we demonstrate high accuracy, with mean errors of under 4% in both slew and delay as compared to HSPICE. We show a speedup of over five orders of magnitude over HSPICE and a speedup of about $92\times$ over conventional CSMs.

I. INTRODUCTION

VARIATIONS in process parameter values and on-chip temperatures have grown larger with shrinking feature sizes. Process variations occur due to phenomena such as proximity effects in photolithography, non-uniform conditions during deposition, and random dopant fluctuations, and lead to fluctuations in parameters such as transistor dimensions, oxide thicknesses, and dopant concentrations [1]–[3]. On-chip temperature variations occur due to power dissipation in the form of heat. Such thermal variations have a significant bearing on the mobilities of electrons and holes, as well as the threshold voltage of the devices. These effects have led to increased shifts in circuit performance, due to which a significant fraction of the total number of acceptable dies may fail to achieve the prescribed performance goals. To overcome this problem, designers must build resilient circuits that meet their performance goals in spite of these variations.

A key enabler for variation-tolerant design is the ability to simulate the timing behavior of a circuit during the design process using static timing analysis (STA). Traditional standard cell modeling approaches represent the delay and output slew as nonlinear functions of the input slew and output

load capacitance [4]. When interconnect resistance became significant, these methods were replaced by the notion of effective capacitance [5]. However, this approach models the input as a saturated ramp with piecewise constant slope, and was further enhanced by the development of CSMs, which represent a cell as a voltage controlled current source and provide fast and accurate timing estimates.

A CSM approach termed as “Blade” [6] represents the cell as a voltage-controlled current source (VCCS) with an internal capacitance and a time-shifted input waveform driving an arbitrary load. A lookup table, indexed by the input voltage, V_{in} , and the output voltage, V_{out} , models the VCCS current, I_{out} . These ideas were further refined in [7]–[12]. The work in [7], [8] removed the assumptions of linearity, and [9]–[12] addressed multiple input switching and stack effects. Further, a current source model based on orthogonal functions was proposed in [13], and an approach based on the small-signal model of a transistor was built in [14].

Within the CSM framework, process variations are commonly captured through the use of process corners. Traditionally, temperature variations were also handled using corner-based methods, but this is no longer viable. Corner-based approaches are predicated on the idea that the timing varies monotonically over the temperature range, but this is no longer the case with thermally-driven variations [15]. In nanometer-scale technologies, elevated temperatures cause reductions in device mobilities (which tend to increase the delay) as well as reductions in threshold voltages (which tend to decrease the delay). The interplay between these effects may cause the circuit delay to increase monotonically (negative temperature dependence), decrease monotonically (positive temperature dependence), or vary nonmonotonically (mixed temperature dependence) with temperature. In the last case, the worst case may occur in the interior of the temperature range, rather than at its edges. As a result, a set of temperature corners is no longer adequate, and circuit delays must be simulated as functions of temperature.

Therefore, a first necessary enhancement of CSMs involves extending them to determine the cell delay as a function of temperature. This capability is useful not only for circuit analysis but also for building optimization techniques that compensate for temperature variations [3], [16]–[19].

A second way in which CSMs require augmentation is in building an ability to simulate cell timing in the presence of body biases. The application of adaptive body biases (ABBs) allows circuits to be made resilient and variation-tolerant by

Manuscript received October 18, 2010; revised August 01, 2011. This work was supported in part by the SRC under contract 2007-TJ-1572 and by the NSF under award CCF-1017778.

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN, 55455, USA (e-mail: saket@umn.edu)

applying a deliberate bias to the body terminals of transistors in a circuit. Realistically, ABB is applied at coarse levels of granularity, e.g., by biasing individual n-wells and/or p-wells, each of which contains a number of transistors. Forward body bias (FBB) effectively reduces the transistor threshold voltage and speeds up the device, at the cost of increased leakage, while reverse body bias (RBB) achieves the opposite effect on speed and leakage. ABB involves the use of FBB or RBB to help dies recover from variations, and may be applied dynamically to tighten the distribution of the dies with maximum operational frequency, while simultaneously meeting the leakage power constraints [1]–[3], [17], [20].

Traditional CSMs simulate the circuit at fixed values of the body bias ($v_{bp} = v_{bn} = 0$) and at fixed values of the temperature. The obvious extensions to existing CSMs that enable them to capture body biases and temperature effects are rather inefficient. In principle, the body terminal of a device can be considered to be another port, and the cell can be accordingly characterized by creating a look-up table for various combinations of body biases, v_{bp}, v_{bn} . Further, such lookup tables would have to be constructed for various temperature values. However, this increases the amount of memory used as well as the characterization time significantly over the zero body bias and the nominal temperature case. For instance, for 10 values each of v_{bp} and v_{bn} , and for 10 values of temperature, the table for each library cell becomes $1000\times$ larger. The need to access a larger lookup table may also result in a significant concomitant increase in the simulation runtime of CSM macromodels.

This paper develops efficient timing characterization methods for building CSMs that incorporate changes in the body bias and the temperature. Since ABB is applied at the granularity of a well, we assume that all PMOS transistors in a cell have the same body bias value, v_{bp} , and all NMOS devices are biased at v_{bn} . Further we assume that all transistors in the cell experience a uniform temperature: this is reasonable, since the rate of decay of temperature with respect to the distance from the cell has a “time constant” that is significantly larger than the size of a cell.

Our framework for incorporating effects of body bias and temperature into the CSM has a very small memory and runtime overhead, while maintaining high levels of accuracy. Our mathematical framework consists of two key steps. First, we intelligently adapt an existing scheme to enable the compaction of look-up tables for the sensitivities of CSM components to body bias and temperature, over the range of allowable values of both the applied body bias and on-chip temperature. Our second key contribution is to develop a novel waveform sensitivity model for evaluating the impact of the applied body bias and variations in temperature, which provides accurate waveforms at the output of the cell under any body bias or temperature condition, with minimal computation. The essential idea of this approach is that since body bias or temperature variation constitutes a small perturbation to the nominal waveform, it should be possible to determine the perturbed waveform cheaply by determining and saving the parameters that compute its shift from the nominal waveform.

We develop a scheme for characterizing this perturbation and computing it efficiently. Specifically, mathematical models for such parameters are developed and further analyzed for their independence over body bias and temperature variations for an efficient computation of such parameters.

The remainder of the paper is organized as follows. Section II presents the development of sensitivity models for CSM components to handle variations in body bias and temperature. Section III presents our algorithms for compacting the CSM sensitivity tables. Then we present the conventional macromodel solvers used in state-of-the-art CSMs in Section IV, and is followed by a description of our method for fast output waveform evaluation in Section V. Section VI presents experimental results on a set of library cells in a 45nm technology. We then present the conclusion of our work in Section VII.

II. CSM SENSITIVITY MODEL DEVELOPMENT

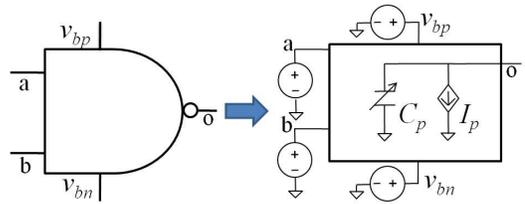


Fig. 1: Example of a CSM: the output port is modeled as a nonlinear VCCS dependent on all input port voltages, in parallel with a nonlinear capacitance.

The CSM is a gate-level black-box abstraction of a cell in a library, with the same input and output ports as the original cell. Our CSM structure, shown in Figure 1, is of the type proposed in [6], and is augmented to model nonlinearities as in [9]. Specifically, output port p is replaced by a nonlinear voltage-controlled-current-source (VCCS), I_p , in parallel with a nonlinear capacitance, C_p . The VCCS model enables the CSM to be load-independent, and permits it to handle an arbitrary electrical waveform at its inputs. The CSM is characterized in terms of the value of I_p and the charge, Q_p , stored on the capacitor, C_p . The variables, I_p and Q_p , are functions of all input and output port voltages and temperature, and are determined by characterizing the cell at various port voltages, body bias combinations and temperatures as follows:

$$I_p = F(V_i, V_o, v_{bp}, v_{bn}, \Delta T) \quad (1)$$

$$Q_p = G(V_i, V_o, v_{bp}, v_{bn}, \Delta T) \quad (2)$$

The parameters I_p and Q_p are modeled using the functions F and G , respectively, and V_i and V_o are, respectively, the voltages at the transitioning input and output ports of the cell. We use the term ΔT to represent the temperature offset from a baseline temperature value, taken here to be room temperature (25°C). In the temperature range of $[-25^\circ\text{C}, 125^\circ\text{C}]$ that we work in, the range for the values of ΔT is $[-50^\circ\text{C}, 100^\circ\text{C}]$.

For a cell, I_p characterization involves DC simulations over multiple combinations of DC values of (V_i, V_o) , while Q_p is characterized through a set of transient simulations [9]. The

presentation of our model is targeted to the more widely-used scenario of single-input switching for gates with a single output, though the idea can easily be extended to multiple input switching (MIS) and multioutput gates, leveraging current work on CSMs on these topics [9], [11], [12].

As mentioned earlier, in order to capture the sensitivity of CSM parameters to the applied body bias and temperature offset, in principle, the circuit could be characterized over a set S of all possible $(v_{bp}, v_{bn}, \Delta T)$ points, treating body terminals as input ports, and temperature offset as the independent variable. Since the allowable values of the applied body biases and temperature offset change in discrete steps, the cardinality of this set is large, and the corresponding characterization would be computationally intensive, even as a precharacterization step that is to be performed once for a technology. Moreover, memory requirements of the table multiply significantly over the current characterization procedure at zero body bias and zero temperature offset.

We observe through simulations that the functions F and G depend *much more* weakly on v_{bp} , v_{bn} , and ΔT as compared to V_i or V_o . Hence, a simpler model can be utilized to save on this computation. We thus develop sensitivity models of CSM with respect to v_{bp} , v_{bn} and ΔT (as we will soon show, the models with respect to body bias and temperature are independent), and then present a scheme to incorporate the effects of the two.

A. Independence of Body Bias and Temperature Effects

Next, we explain the rationale for analyzing the effects of body bias and temperature independently. Body bias (a change in the substrate bias voltage, V_{BS}) changes the threshold voltage, V_{th} . The sensitivity of V_{th} with respect to V_{BS} can be captured from following equation [21]:

$$\frac{\partial V_{th}}{\partial V_{BS}} = \frac{C_{dep}}{C_{ox}} \quad (3)$$

where C_{dep} is the depletion capacitance of the MOS transistor and C_{ox} is the oxide capacitance. C_{dep} is a very weak function of temperature, being proportional to the inverse square root of the built-in potential. Similarly, it is observed that the expression for V_{th} sensitivity with respect to temperature is independent of V_{BS} [21]. Hence, the effects of changes in body bias and temperature on MOS transistors can be treated as independent. Since I_p and Q_p essentially abstract the internal cell behavior, the effects of body bias and changes in temperature on I_p and Q_p can also be assumed to be independent. This is further verified by the model formulations and accuracy results as presented in the subsequent subsections and sections.

B. CSM Body Bias Sensitivity Model

We first present the body bias sensitivity model, which is independent of the changes in temperature and constructed at $\Delta T = 0^\circ\text{C}$ (i.e., at room temperature). We construct a polynomial approximation for the variations of I_p and Q_p with respect to (v_{bp}, v_{bn}) . Our simulations show that a linear approximation yields an average of 2.0% relative error with respect to HSPICE, evaluated over all (v_{bp}, v_{bn}) points, for

all (V_i, V_o) points. The CSM is now modified by using the equations:

$$I_p(V_i, V_o, v_{bp}, v_{bn}, 0) = I_p^Z \cdot (1 + a_I(V_i, V_o)v_{bp} + b_I(V_i, V_o)v_{bn}) \quad (4)$$

$$Q_p(V_i, V_o, v_{bp}, v_{bn}, 0) = Q_p^Z \cdot (1 + a_Q(V_i, V_o)v_{bp} + b_Q(V_i, V_o)v_{bn}) \quad (5)$$

where $I_p^Z = F(V_i, V_o, 0, 0, 0)$, $Q_p^Z = G(V_i, V_o, 0, 0, 0)$, and $\{a_I, b_I, a_Q, b_Q\}$ correspond to the sensitivity of the function to the corresponding body bias. These parameters are characterized at a discrete set of (V_i, V_o) values and are saved in a lookup table.

The characterization of I_p and Q_p using equations (4) and (5) can now be carried out using a minimum of three simulations at each (V_i, V_o) , since it is a linear model; however, additional redundancy is preferable to account for the small nonlinearities, and a linear least squares fit can be used instead.

For notational simplicity, we will define the following functions:

$$L_I(v_{bp}, v_{bn}) = 1 + a_I(V_i, V_o)v_{bp} + b_I(V_i, V_o)v_{bn} \quad (6)$$

$$L_Q(v_{bp}, v_{bn}) = 1 + a_Q(V_i, V_o)v_{bp} + b_Q(V_i, V_o)v_{bn} \quad (7)$$

Clearly,

$$I_p(V_i, V_o, v_{bp}, v_{bn}, 0) = I_p^Z L_I(v_{bp}, v_{bn}) \quad (8)$$

$$Q_p(V_i, V_o, v_{bp}, v_{bn}, 0) = Q_p^Z L_Q(v_{bp}, v_{bn}) \quad (9)$$

C. CSM Temperature Sensitivity Model

We now construct the temperature sensitivity model at zero body bias. We observe that the variations of I_p and Q_p with ΔT are nonlinear, unlike the body bias case where a linear approximation was adequate. We employ a second-order polynomial approximation, and find that the fit has an average relative error of 1.6% relative error in comparison with HSPICE simulations. The CSM for the temperature sensitivity model with the first and second order sensitivities in temperature offset is now represented by the following modified equations:

$$I_p(V_i, V_o, 0, 0, \Delta T) = I_p^Z \cdot (1 + c_I(V_i, V_o)\Delta T + r_I(V_i, V_o)\Delta T^2) \quad (10)$$

$$Q_p(V_i, V_o, 0, 0, \Delta T) = Q_p^Z \cdot (1 + c_Q(V_i, V_o)\Delta T + r_Q(V_i, V_o)\Delta T^2) \quad (11)$$

where I_p^Z, Q_p^Z are as defined above, and $\{c_I, c_Q, r_I, r_Q\}$ correspond to the sensitivity of the function to the corresponding powers of the temperature offset, ΔT . As in the case of $\{a_I, b_I, a_Q, b_Q\}$, these parameters are characterized at a discrete set of (V_i, V_o) values and saved in a lookup table. Since the temperature sensitivity model is a second order model, we need at least three points to determine the values of $\{c_I, r_I, c_Q, r_Q\}$.

As before, for notational simplicity, we will define the following functions:

$$S_I(\Delta T) = 1 + c_I(V_i, V_o)\Delta T + r_I(V_i, V_o)\Delta T^2 \quad (12)$$

$$S_Q(\Delta T) = 1 + c_Q(V_i, V_o)\Delta T + r_Q(V_i, V_o)\Delta T^2 \quad (13)$$

Clearly,

$$I_p(V_i, V_o, 0, 0, \Delta T) = I_p^Z S_I(\Delta T) \quad (14)$$

$$Q_p(V_i, V_o, 0, 0, \Delta T) = Q_p^Z S_Q(\Delta T) \quad (15)$$

D. CSM Complete Sensitivity Model

The complete body bias and temperature sensitivity model can now be formulated by integrating the models of I_p and Q_p with body bias from equations (8), (9), and with temperature offset from equations (14), (15). The complete model is constructed as follows:

$$I_p(V_i, V_o, v_{bp}, v_{bn}, \Delta T) = I_p^Z \cdot L_I(v_{bp}, v_{bn}) \cdot S_I(\Delta T) \quad (16)$$

$$Q_p(V_i, V_o, v_{bp}, v_{bn}, \Delta T) = Q_p^Z \cdot L_Q(v_{bp}, v_{bn}) \cdot S_Q(\Delta T) \quad (17)$$

Simulations show that the above model yields approximations with an average of 2.9% relative error with respect to HSPICE. This also justifies our assumption that the effects of body bias and changes in temperature on CSM components can be analyzed independently.

III. COMPACT CSM FORMULATION

As described in Sections I and II, the lookup tables obtained for $\{a_I, b_I, a_Q, b_Q\}$ and $\{c_I, r_I, c_Q, r_Q\}$ reduce the excessive memory requirements described in Section I. However, we still need a separate lookup table (indexed by (V_i, V_o)) for each parameter of every cell in the library. If we can further reduce the size of these tables by suitably compacting them, we can gain more in terms of memory overheads induced. We thus present the development of a compact lookup table scheme used for reducing the size of such lookup tables.

A. Table Size Reduction for Conventional CSMs

As a preliminary step, we attempt to apply the method in [22] to create compact lookup tables for I_p and Q_p for the zero body bias and nominal temperature case, i.e., I_p^Z and Q_p^Z , with controlled loss of accuracy. For general values of the body bias and temperature, we must also create lookup tables for $\{a_I, b_I, a_Q, b_Q\}$ and $\{c_I, r_I, c_Q, r_Q\}$ at each value of (V_i, V_o) : as we will see, for these parameters, a direct extension of the method in [22] does not yield satisfactory results.

We first overview the procedure in [22]. This method begins with an $n \times n$ table of characterized points, indexed by variables x and y in the horizontal and vertical directions, respectively. The idea behind table size reduction is to keep a subset of all these points and to interpolate the rest. For instance, consider the rectangle bounded by points (x_1, y_1) , (x_2, y_1) , (x_2, y_2) , and (x_1, y_2) : a point (x, y) within this rectangle can be dropped if the interpolation error in its value, using these points lies within a specified bound.

Instead of an expensive enumeration, the work in [22] presents a dynamic programming method for reducing a two-dimensional $n \times n$ table. The objective of the algorithm is to create a smaller $m \times m$ table, where m is prespecified, while minimizing the total error corresponding to the points that are dropped from the table. The procedure begins by constructing

an initial 2×2 table corresponding to the points at the four corners of the table. Next, this table is expanded to include additional entries using the idea of h -hops.

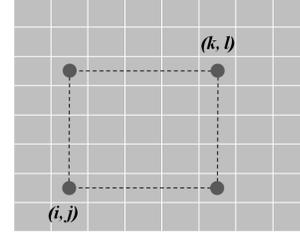


Fig. 2: The initial step, considering all rectangles from any point (i, j) , extending to any point (k, l) at the northeast corner.

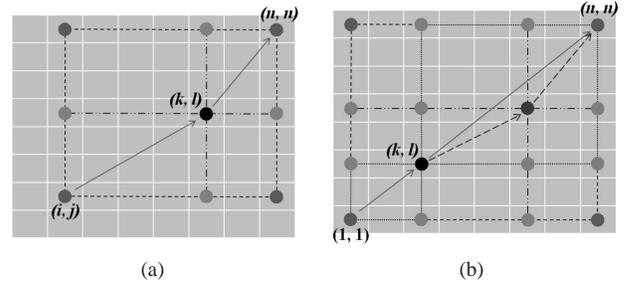


Fig. 3: (a) A 1-hop solution from (i, j) to (n, n) , through an intermediate point, (k, l) . (b) A 2-hop solution from $(1, 1)$ to (n, n) through an intermediate point, (k, l) uses a previously computed optimal 1-hop solution from (k, l) to (n, n) .

In the initial step, we consider all rectangles originating at a point (i, j) at the southwest corner, extending to any point (k, l) at the northeast corner, as shown in Figure 2. We compute the error metric over the rectangle, corresponding to the case where only the points at the four corners of the rectangle are kept in the lookup table, and all internal points are dropped. The error metric is the sum of the interpolation errors for all points within and on the perimeter of the rectangle. Each such rectangle corresponds to an optimal substructure for dynamic programming: the optimal solution will be composed from some (but not all) such substructures.

Next, we define a 1-hop operation. We optimize the region bounded by point (i, j) to the southwest and (n, n) to the northeast by finding an optimal point (k, l) within this region. Here, optimality is defined as follows: the point (k, l) divides the region into four subregions, as shown in Figure 3(a), and over all candidate (k, l) points, the optimal point minimizes the total error summed up over these four subregions. Since the error over each rectangle was calculated in the initial step, this step involves enumerating all candidate (k, l) points, and summing up the previously calculated error over the rectangles in constant time for each such point. We refer to this as a 1-hop, indicating that for each (i, j) , the table “hops” over a single point, corresponding to the optimal (k, l) , on the way to (n, n) . The associated optimal error encountered is the 1-hop error for (i, j) .

In general, an h -hop from (i, j) to (n, n) finds a point (k, l) such that the error from (i, j) to (k, l) , plus the $(h - 1)$ -hop error from (k, l) to (n, n) , is minimized over all candidate

points (k, l) . To obtain an $m \times m$ table, the procedure stops after $m - 1$ hops, and the optimal m -hop from $(1, 1)$ to (n, n) provides the compact table. Figure 3(b) shows an example of a 2-hop solution from $P(1, 1)$ to $P(n, n)$; if the algorithm were to stop here, it would result in a 4×4 compacted table. The computational complexity of this algorithm is $O(m \cdot n^4)$, but as n is typically small ($n = 30$ in our simulations), this remains tractable, as we will show in Section VI-A that the runtimes for this scheme are reasonable.

It should be noted that although this method proceeds along the main diagonal of the table (in the north-east direction), the interpolation error is computed by considering all the four end points of a rectangle (in Figure 3(a) for instance). Thus, it also considers the interpolation error induced along the other diagonal, and the rows and columns of the table as well. While this method is not exact (for example, for an h -hop, it does not entertain the possibility of an $(h - 1)$ hop to (k, l) and then a 1-hop to (n, n)), in practice it is seen to work well. A faster version of the algorithm, which trades off accuracy for speed, is also proposed in [22].

B. Modifications for Sensitivity Tables

As stated earlier, the above approach works well for characterizing I_p and Q_p , where neighboring entries have similar magnitudes. However, in case of the sensitivity parameters, $\{a_I, b_I, a_Q, b_Q\}$ and $\{c_I, r_I, c_Q, r_Q\}$, there can be large differences in the values of neighboring parameters. This is illustrated in Figure 4(a) and (b), which show, respectively, the values of $a_Q = \partial Q_p / \partial v_{bp}$ and $c_I = \partial I_p / \partial \Delta T$ for an inverter cell¹. Large “outliers” (i.e., values of large magnitude) are clearly visible on the plot.

The presence of these outliers is attributed to the nature of variation of the values of I_p^Z and Q_p^Z with (V_i, V_o) , and the way these values are derived in the CSM. At a particular (V_i, V_o) bias point, it is quite possible that only a small current flows inside the input and output terminals of the cell. Since the magnitudes of these inflowing currents decide the values of I_p^Z and Q_p^Z , the values of resultant I_p^Z and Q_p^Z are also small. Hence, any change relative to this small value becomes large and is reflected as a large sensitivity value.

In principle, since these I_p^Z and Q_p^Z values are small, we may consider setting the corresponding sensitivities to zero. This, however, has been observed to create inaccuracies in waveform evaluations (the waveform evaluation techniques are described in Sections IV and V) for when such changes are multiplied by other quantities with relatively higher magnitudes (temperature offset for instance), the net contribution from these small changes, to the computed values of $V_o(t)$ or of the waveform sensitivities, becomes significant, and hence cannot be neglected.

For such data, it can easily be shown that the approach in [22], which depends on gridding the table in the coordinate directions, is poorly compacted, i.e., the interpolation errors in the reduced table are large. Such errors are demonstrated to be easily visible in the output response, where they appear as

“kinks” in the CSM-based waveform that do not exist in the corresponding HSPICE waveform. This happens due to the fact that an interpolation error caused by the presence of these outliers causes an error in I_p and Q_p values, which causes the solver (described in the next section) to generate errors in output waveforms. A sample waveform with the use of compacted I_p and Q_p tables, as generated by the solver for a rising input ramp is shown in Figure 5. As is seen, due to poor compaction, kinks appear in the evaluated waveform. The incorrect waveform also incurs slew and delay errors.

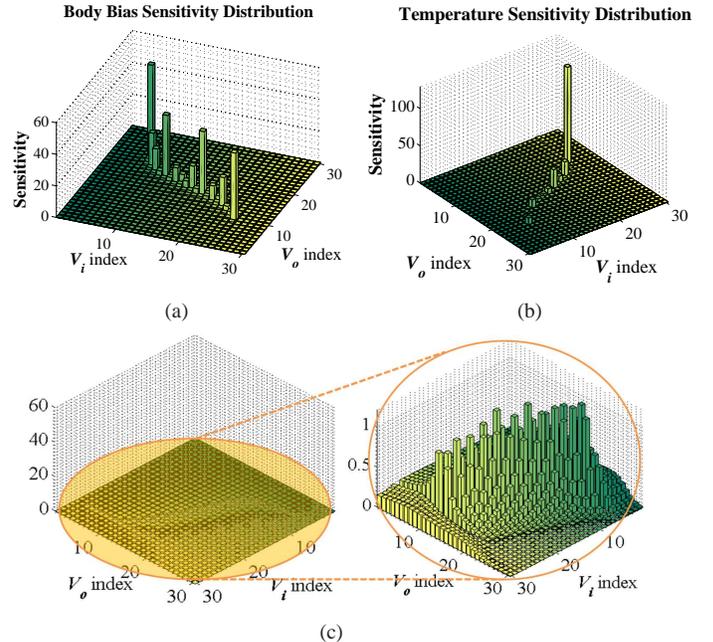


Fig. 4: The CSM sensitivity parameter distribution for (a) a_Q and (b) c_I as functions of (V_i, V_o) . (c) The resultant lookup table for a_Q , when all the outliers have been removed and saved separately in a table.

We propose a simple method for avoiding these problems, based on the observation that for these sensitivity parameters, such outliers are few in number and have relatively large magnitudes. We therefore tabulate and save the outliers separately. As can be observed from Figure 4(a) and (b), the number of outliers is quite small compared to the total number of data points. Thus, a separate tabulation of outliers would incur negligible overhead.

In order to tabulate the outliers separately, given the set of all points, we find the mean and variance over all entries. Any entry that is over k variances from the mean is found to be an outlier; in practice, we find $k = 2$ to be an adequate value. The removed entry at table location (x, y) is then replaced by a dummy point, the error contribution (to the total error) from which is zero. The modified table is then compacted using the algorithm in Section III-A.

When a table entry is requested, we first determine whether the accessed point is an outlier: if so, we fetch it from the outliers list; else, we find it using the compacted look-up table.

With the outliers separated, the variations in remaining lookup table become more uniform. Table I shows the list of

¹Similar behavior is seen for $a_I, b_I, b_Q, r_I, c_Q,$ and r_Q .

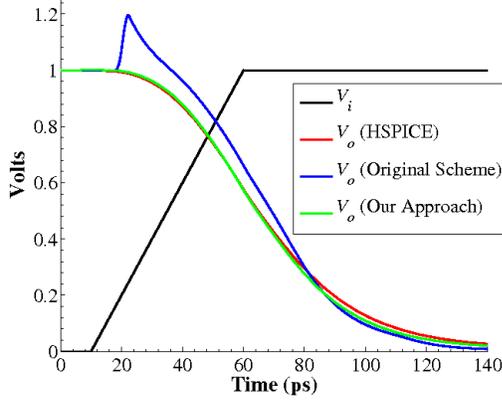


Fig. 5: The presence of outliers yields poor compaction of the lookup tables when the original scheme from [22] is used. This results in incorrectly evaluated output waveforms with kinks at some time points. Our approach however, with a mechanism for separation of outliers, results in the correctly evaluated output waveform with minimal errors.

separately tabulated outliers for a lookup table for a_Q . Further, Figure 4(c) shows the remaining entries for a_Q in the 2-D lookup table indexed by V_i, V_o . As is clearly seen, the removal of outliers make the variation in the lookup table more uniform, allowing for a high compaction using the original algorithm.

TABLE I: The outlier table for a_Q

V_i index	V_o index	a_Q
10	25	42.6
13	22	18.9
16	16	15.5
.	.	.
.	.	.
22	7	60.7

This method of separating the outliers removes the kinks present in the $V_o(t)$ waveforms. As shown in Figure 5, the smooth waveform obtained from the solver using our approach is no longer characterized by kinks, as compared to the waveform which had kinks due to the errors caused by original compaction scheme. The waveform using our approach further has negligible slew and delay errors.

A potential alternative for dealing with such outliers is to decrease the size of (V_i, V_o) voltages steps at which I_p^Z and Q_p^Z are characterized, making the variation of sensitivities more uniform. We observe that this requires us to increase the value of n by about 6-9 \times for different tables, resulting in a large increase in the storage space required. For a small number of outliers, this posed as a significant increase in the memory requirements for a library with different cells. It also prohibitively increases the computational time of the compression algorithm ($\propto n^4$). Therefore, an intermediate approach of saving outliers separately keeps both the storage space and the compression time tractable.

IV. THE MACROMODEL SOLVER

Using the approaches described so far, the cell library is characterized to determine the I_p and Q_p characterization

tables at zero body bias and zero temperature offset, and the compressed CSM sensitivity parameter tables for the body bias coefficients $\{a_I, b_I, a_Q, b_Q\}$ and the temperature offset coefficients $\{c_I, r_I, c_Q, r_Q\}$.

A. Using the Macromodel in a Solver

To solve the case of a gate driving an interconnect, including cases that involve coupled lines and crosstalk, it is enough to consider the situation where a gate drives a load described by an RC π -model as shown in Figure 6. Standard techniques such as the O'Brien-Savarino approach [23] are used in our work to reduce an arbitrary interconnect load to a π -model at the driving point. We first obtain the waveform at the driving point node V_o , and then we evaluate the waveform at any sink node in the RC network by solving a linear system using standard model order reduction methods.

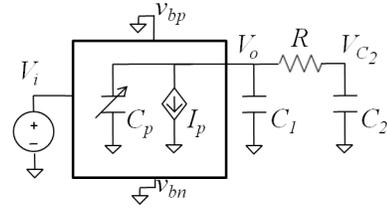


Fig. 6: A CSM for a gate, under zero body bias and zero temperature offset, driving a π load.

We analyze the case of a gate output driving a π -load in the absence of body bias and at zero temperature offset, as shown in Figure 6. Finding the output voltage waveform involves solving the equation:

$$I_p^Z + I_{Q_p}^Z = I_{C_1} + I_{C_2} \quad (18)$$

$$\text{where } I_{Q_p}^Z = \frac{dQ_p^Z}{dt}$$

$$I_{C_1} = C_1 \frac{dV_o}{dt}$$

$$I_{C_2} = C_2 \frac{dV_{C_2}}{dt}$$

$$I_p^Z = F(V_i, V_o, 0, 0, 0)$$

$$Q_p^Z = G(V_i, V_o, 0, 0, 0)$$

Equation (18) is a nonlinear differential equation in $V_o(t)$, and the input voltage, $V_i(t)$, is known. This equation can be solved using routine circuit simulation methods. We apply the Backward Euler formula to Q_p, V_o and V_{C_2} with a time step h , going from time n to time $n+1$ (the superscript $n+1$ is dropped for notational simplicity) to get:

$$Q_p = Q_p^n + hI_{Q_p} \quad (19)$$

$$C_1 V_o = C_1 V_o^n + hI_{C_1} \quad (20)$$

$$C_2 V_{C_2} = C_2 V_{C_2}^n + hI_{C_2} \quad (21)$$

Moreover, using Ohm's Law, we have $V_{C_2} = V_o - RI_{C_2}$. Substituting V_{C_2} from this in equation (21), we have:

$$I_{C_2} = \frac{C_2(V_o - V_{C_2}^n)}{h + RC_2} \quad (22)$$

We then obtain the values of I_{Q_p} from equation (19), of I_{C_1} from equation (20) and of I_{C_2} from equation (22), and substitute them in equation (18) to obtain:

$$I_p + \frac{Q_p - Q_p^n}{h} = \frac{C_1(V_o - V_o^n)}{h} + \frac{C_2(V_o - V_{C_2}^n)}{h + RC_2}$$

Solving this for V_o , we arrive at the following expressions:

$$V_o = \frac{1}{A} [hC_2V_{C_2}^n + B(C_1V_o^n + hI_p + Q_p - Q_p^n)] \quad (23)$$

$$\text{where } A = (hC_1 + hC_2 + RC_1C_2) \quad (24)$$

$$B = (h + RC_2) \quad (25)$$

Obtaining V_o , we substitute I_{C_2} from equation (22) in equation (21) to solve for V_{C_2} :

$$V_{C_2} = \frac{1}{B} [hV_o + RC_2V_{C_2}^n] \quad (26)$$

Thus we have obtained the expressions for both the unknown port voltages in terms of known quantities. However, such expressions are still implicit, and hence must be solved iteratively.

B. Newton-Raphson Solver

The approach conventionally employed in CSM solvers is to solve the nonlinear equation (23), through iterative Newton-Raphson linearization. This approach is hereby termed as the Newton-Raphson Solver, and referred as such in the rest of the sections. In the $(k+1)^{\text{th}}$ iteration, we use the k^{th} iteration value, shown by the additional subscript k , to obtain:

$$\begin{aligned} AV_o = BC_1V_o^n + hC_2V_{C_2}^n + hB \left(I_{p,k} + \frac{\partial I_p}{\partial V_o} \Big|_k (V_o - V_{o,k}) \right) \\ + B \left(Q_{p,k} + \frac{\partial Q_p}{\partial V_o} \Big|_k (V_o - V_{o,k}) - Q_p^n \right) \\ V_o = V_{o,k} - \frac{AV_{o,k} - hC_2V_{C_2}^n - B(C_1V_o^n + hI_{p,k} + Q_{p,k} - Q_p^n)}{A - B(h \frac{\partial I_p}{\partial V_o} \Big|_k + \frac{\partial Q_p}{\partial V_o} \Big|_k)} \end{aligned} \quad (27)$$

This computation is carried out by references to the look-up tables for I_p and Q_p , with the appropriate use of interpolation as necessary, and the use of finite differences to compute derivatives.

V. FORMULATION OF WAVEFORM SENSITIVITY MODEL

The Newton-Raphson solver in Section IV-B forms the basis for a procedure for computing the waveform under any body bias and temperature condition using conventional CSM solvers. However, evaluation of the delays and slews of the gates under numerous body bias and temperature offset conditions entails multiple simulations of the *entire* output voltage waveform at each combination of body bias and temperature value. Applications that require timing analysis at multiple body biases and at multiple temperature values include [1]–[3], [17], [19].

Intuitively, the repeated computation of full waveforms from scratch seems unnecessarily excessive, for several reasons. First, the application of body bias or a variation in temperature

corresponds to a perturbation to a base case, such as the zero body bias and zero temperature offset case, and it should be possible to compute the waveform at nonzero body bias and temperature offset based on the zero body bias and zero temperature offset case, with some consideration of body bias and temperature sensitivities, much more cheaply than the above procedure. Second, as discussed and shown before in Section II, the effects of changes in body bias and temperature on CSM can be decoupled. Thus it should be possible to decouple and independently compute the effects of body bias and temperature changes on the output waveforms too. Third, in most cases, designers are interested not in the entire waveform, but specific properties of the gate output, such as its delay and output transition time. In this section, we demonstrate the efficient computation of such metrics under changing body bias and temperature without the need for numerous table look-up operations.

A. Waveform Sensitivity Models

Consider the case when we have the cell maintained at zero temperature offset ($\Delta T = 0^\circ\text{C}$), but with a nonzero applied body bias (v_{bp}, v_{bn}). For various values of (v_{bp}, v_{bn}) , the solution of the waveform under the framework of equation (27) entails multiple accesses to the look-up tables for I_p and Q_p . The entries that are accessed in these tables change according to the applied body bias. However, since body bias is a small perturbation, in practice, the accessed entries in each table at each step of the algorithm are relatively close to each other, and can be viewed as perturbations to a nominal case.

Therefore, we propose to capture the output waveform at zero temperature offset for nonzero body bias case as a perturbation to the waveform with zero body bias and zero temperature offset as follows:

$$V_o(t) = V_o^Z(t) + \alpha(v_{bp}, v_{bn}, t) \cdot v_{bp} + \beta(v_{bp}, v_{bn}, t) \cdot v_{bn} \quad (28)$$

where $V_o^Z(t)$ represents the output waveform, $V_o(t)$, with zero body bias and zero temperature offset, and $\alpha(v_{bp}, v_{bn}, t)$ and $\beta(v_{bp}, v_{bn}, t)$ are time-varying body bias perturbation parameters that are precisely defined as:

$$\begin{aligned} \alpha(v_{bp}, v_{bn}, t) &= \frac{\partial V_o(t)}{\partial v_{bp}} \\ \beta(v_{bp}, v_{bn}, t) &= \frac{\partial V_o(t)}{\partial v_{bn}} \end{aligned} \quad (29)$$

Similarly, if we consider the variation in temperature of the cell, the cell being maintained at zero body bias, we can formulate a linear model as above for capturing the output waveform at any temperature (with a nonzero temperature offset, ΔT) as perturbation to the output waveform at nominal temperature (with zero temperature offset ΔT):

$$V_o(t) = V_o^Z(t) + \sigma(\Delta T, t) \cdot \Delta T \quad (30)$$

where $V_o^Z(t)$ is as as described above, and $\sigma(\Delta T, t)$ is time-varying temperature perturbation parameter that is precisely defined as:

$$\sigma(\Delta T, t) = \frac{\partial V_o(t)}{\partial \Delta T} \quad (31)$$

The following two results provide a precise formula for $\alpha(v_{bp}, v_{bn}, t)$, $\beta(v_{bp}, v_{bn}, t)$ and $\sigma(\Delta T, t)$. We first present the results (proved in the Appendix), and then discuss how the computational cost of evaluating these quantities can be significantly reduced.

Theorem 1 *The waveform sensitivity parameters from equation (28), $\alpha(v_{bp}, v_{bn}, t)$ and $\beta(v_{bp}, v_{bn}, t)$, are given by:*

$$\alpha(v_{bp}, v_{bn}, t) = N_\alpha / D_{\alpha,\beta} \quad (32)$$

$$\beta(v_{bp}, v_{bn}, t) = N_\beta / D_{\alpha,\beta} \quad (33)$$

$$N_\alpha = B \left[\alpha^n C_1 + h a_I I_p^Z + a_Q Q_p^Z - a_Q^n Q_p^{Z,n} - Q_p^{Z,n} \frac{\partial a_Q^n}{\partial V_o^n} \alpha^n v_{bp} - Q_p^{Z,n} \frac{\partial b_Q^n}{\partial V_o^n} \alpha^n v_{bn} - \frac{\partial Q_p^{Z,n}}{\partial v_{bp}} L_Q^n(v_{bp}, v_{bn}) \right] + h C_2 \frac{\partial V_{C_2}^n}{\partial v_{bp}},$$

$$N_\beta = B \left[\beta^n C_1 + h b_I I_p^Z + b_Q Q_p^Z - b_Q^n Q_p^{Z,n} - Q_p^{Z,n} \frac{\partial a_Q^n}{\partial V_o^n} \beta^n v_{bp} - Q_p^{Z,n} \frac{\partial b_Q^n}{\partial V_o^n} \beta^n v_{bn} - \frac{\partial Q_p^{Z,n}}{\partial v_{bn}} L_Q^n(v_{bp}, v_{bn}) \right] + h C_2 \frac{\partial V_{C_2}^n}{\partial v_{bn}},$$

$$D_{\alpha,\beta} = (-B) \left[h I_p^Z \left(\frac{\partial a_I}{\partial V_o} v_{bp} + \frac{\partial b_I}{\partial V_o} v_{bn} \right) + h \frac{\partial I_p^Z}{\partial V_o} L_I(v_{bp}, v_{bn}) + Q_p^Z \left(\frac{\partial a_Q}{\partial V_o} v_{bp} + \frac{\partial b_Q}{\partial V_o} v_{bn} \right) + \frac{\partial Q_p^Z}{\partial v_{bp}} L_Q(v_{bp}, v_{bn}) \right] + A$$

where terms using the superscript n are understood to correspond to their values at the previous (n^{th}) time step, and the superscript Z refers to the case where $v_{bp} = v_{bn} = 0\text{V}$, and $\Delta T = 0^\circ\text{C}$.

Theorem 2 *The waveform temperature sensitivity parameter from equation (30), $\sigma(\Delta T, t)$, is given by:*

$$\sigma(\Delta T, t) = N_\sigma / D_\sigma \quad (34)$$

$$N_\sigma = B \left[\sigma^n C_1 + h(c_I + 2r_I \Delta T) I_p^Z + (c_Q + 2r_Q \Delta T) Q_p^Z - (c_Q^n + 2r_Q^n \Delta T) Q_p^{Z,n} - \frac{\partial Q_p^{Z,n}}{\partial \Delta T} S_Q^n(\Delta T) - Q_p^{Z,n} \frac{\partial c_Q^n}{\partial V_o^n} \sigma^n \Delta T - Q_p^{Z,n} \frac{\partial r_Q^n}{\partial V_o^n} \sigma^n \Delta T^2 \right] + h C_2 \frac{\partial V_{C_2}^n}{\partial \Delta T},$$

$$D_\sigma = A - B \left[\left(h I_p^Z \frac{\partial c_I}{\partial V_o} + Q_p^Z \frac{\partial c_Q}{\partial V_o} \right) \Delta T + h \frac{\partial I_p^Z}{\partial \Delta T} S_I(\Delta T) + \left(h I_p^Z \frac{\partial r_I}{\partial V_o} + Q_p^Z \frac{\partial r_Q}{\partial V_o} \right) \Delta T^2 + \frac{\partial Q_p^Z}{\partial \Delta T} S_Q(\Delta T) \right]$$

where the terms have the notations as described above.

Theorems 1 and 2 enable the efficient computation $V_o(t)$ at any body bias value and temperature offset using a closed form expression, dependent only on the values of V_o at previous time steps and the values in the waveform at zero body bias and nominal temperature. As a result, the waveform at arbitrary body bias and temperature values can be reproduced if the values of $\alpha(t)$, $\beta(t)$ and $\sigma(t)$ are computed.

B. Simplified Waveform Sensitivity Models

Further simplifications are possible with both the models discussed above. Consider first the body bias model. On investigating dependency of the output waveform on (v_{bp}, v_{bn}) and on $\alpha(v_{bp}, v_{bn}, t)$, $\beta(v_{bp}, v_{bn}, t)$, we observe that:

- 1) The variation in $V_o(t)$ over (v_{bp}, v_{bn}) is nearly linear at each time point of the waveform. Empirically, this can be seen in Figure 7, which shows typical cases for the variation of $V_o(t)$ over (v_{bp}, v_{bn}) for various time points of simulation. This behavior is observed for multiple test cases, and indicates that $\alpha(v_{bp}, v_{bn}, t)$, $\beta(v_{bp}, v_{bn}, t)$ are actually independent of the applied body bias, and are only dependent on t .
- 2) Figure 8 shows the variations in $\alpha(v_{bp}, v_{bn}, t)$ and $\beta(v_{bp}, v_{bn}, t)$ with (v_{bp}, v_{bn}) . The magnitude of these variations were observed to be a maximum of 0.1 for all test cases. Since these parameters are further multiplied by v_{bp} or $v_{bn} \in [-0.3\text{V}, 0.3\text{V}]$ in equation (28), their effects on $V_o(t)$ are expected to be negligible. This is further validated in Section VI.

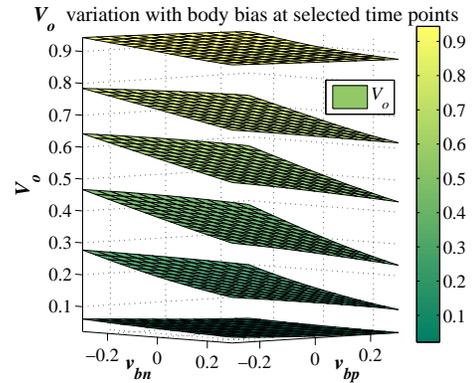


Fig. 7: Typical surface plots for V_o showing the linear nature of V_o variations with (v_{bp}, v_{bn}) , with each surface corresponding to a randomly selected time point during the simulation.

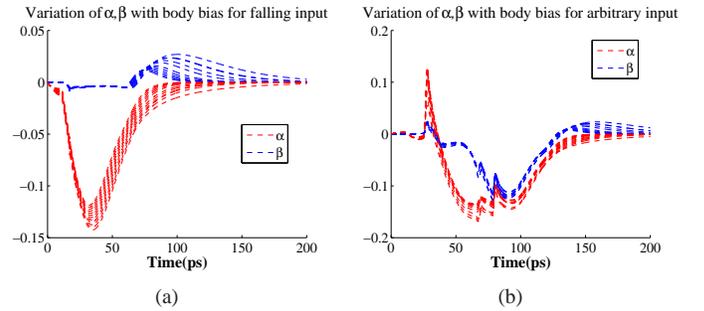


Fig. 8: Simulations showing the variation of $\alpha(t)$ and $\beta(t)$ at a range of body biases from the minimum to the maximum, including zero. Two such test cases are shown in Figure (a) and (b).

This leads to the following approximation, which provides accurate waveforms with very low errors, as demonstrated in Section VI:

$$\alpha(v_{bp}, v_{bn}, t) \approx \alpha_0(t) = \alpha(v_{bp} = 0, v_{bn} = 0, t)$$

$$\beta(v_{bp}, v_{bn}, t) \approx \beta_0(t) = \beta(v_{bp} = 0, v_{bn} = 0, t) \quad (35)$$

The simplified body bias waveform sensitivity model is thus given as follows:

$$V_o(t) = V_o^Z(t) + \alpha_0(t) \cdot v_{bp} + \beta_0(t) \cdot v_{bn} \quad (36)$$

Note that this dramatically reduces the storage requirements for the lookup table. At each time point, this method requires just two additional parameters, α_0 and β_0 .

In order to develop a simplified model with changes in temperature as was done in the body bias case, we investigated the possibility of being able to generate a simplified temperature waveform sensitivity model too. We however find that unlike the body bias case, the following approximation:

$$\sigma(\Delta T, t) \approx \sigma_0(t) = \sigma(\Delta T = 0, t) \quad (37)$$

does not work very well with the temperature waveform sensitivity model. The inaccuracies in the resultant delays and slews, as compared to HSPICE, reach up to 20%. This can be attributed to the nonlinear effects of temperature on the circuit responses, which lead to reduced accuracy when a linear model is used.

Therefore, we apply a more accurate piecewise linear model to address the above inaccuracies. We observe that increasing the value of $|\Delta T|$ increases the inaccuracies in waveform evaluation, and that the magnitude of such errors are not large for smaller values of $|\Delta T|$. Thus, instead of the very simplistic linear approximation as in equation (37), we propose a more accurate and less approximate linear simplified temperature sensitivity model as follows:

$$\begin{aligned} \sigma(\Delta T, t) &\approx \sigma_p(\Delta T, t) \\ &= \sigma(\Delta T = \Delta T_1, t), \Delta T_{MIN} \leq \Delta T < \Delta T_{MIN} + \left\lceil \frac{\Delta T_R}{3} \right\rceil \\ &= \sigma(\Delta T = \Delta T_2, t), \Delta T_{MIN} + \left\lceil \frac{\Delta T_R}{3} \right\rceil \leq \Delta T < \Delta T_{MIN} + \left\lceil \frac{2\Delta T_R}{3} \right\rceil \\ &= \sigma(\Delta T = \Delta T_3, t), \Delta T_{MIN} + \left\lceil \frac{2\Delta T_R}{3} \right\rceil \leq \Delta T \leq \Delta T_{MAX} \end{aligned} \quad (38)$$

where,

ΔT_{MIN} = Minimum value of temperature offset in the range of ΔT

ΔT_{MAX} = Maximum value of temperature offset in the range of ΔT

$$\Delta T_R = \Delta T_{MAX} - \Delta T_{MIN}$$

$$\Delta T_1 = \Delta T_{MIN} + \left\lceil \frac{\Delta T_R}{6} \right\rceil$$

$$\Delta T_2 = \Delta T_{MIN} + \left\lceil \frac{3\Delta T_R}{6} \right\rceil$$

$$\Delta T_3 = \Delta T_{MIN} + \left\lceil \frac{5\Delta T_R}{6} \right\rceil$$

As stated in Section II, the values of ΔT_{MAX} and ΔT_{MIN} are taken to be -50°C and 100°C , respectively. The above formulation in equation (38) states that this temperature range is divided into three ranges of nearly equal size. The waveforms of $\sigma_p(\Delta T, t)$, with ΔT chosen as the central value in each of these intervals, are then evaluated and saved. Although in principle, a waveform corresponding to each of the 13 ΔT values can be saved (giving us the lowest error in the model), a designer would like to save and work with minimal number of waveforms, without losing much in accuracy. We have found through simulations that a choice of 3 different waveform

(described through equation (38)) serves this purpose. As we will show in Section VI, such a choice still preserves the high accuracy. The gain in storage and waveform evaluation speedup on the other hand, is significant.

In other words, instead of computing and saving $\sigma(\Delta T, t)$ at just one temperature point (as in the linear case in equation (37)), we now save the values of $\sigma(\Delta T, t)$ at three distinct values of temperature to provide a better approximation that captures thermal nonlinearities. The simplified temperature waveform sensitivity model is thus given as follows:

$$V_o(t) = V_o^Z(t) + \sigma_p(\Delta T, t) \cdot \Delta T \quad (39)$$

where $\sigma_p(\Delta T, t)$ is given by equation (38). As will be shown in Section VI, the above model yields accurate waveforms for all temperature points.

C. Complete Waveform Sensitivity Model

We now propose the complete body bias and temperature waveform sensitivity model as follows:

$$\begin{aligned} V_o(t) &= V_o^Z(t) \\ &+ \alpha_0(t) \cdot v_{bp} + \beta_0(t) \cdot v_{bn} + \sigma_p(\Delta T, t) \cdot \Delta T \end{aligned} \quad (40)$$

This model is a linear combination of the simplified waveform sensitivity models as given in equations (36) and (39). Note that such a linear combination is possible since the effects of body bias and temperature are independent of each other, as has been discussed in Section II. Equation (40) predicts that the effects of perturbations inside a cell caused due to changes in body bias and temperature, can be captured through a simple linear model of the output voltage in terms of the changes in the body bias and temperature.

To summarize, evaluating the output at b body bias points each for v_{bp} and v_{bn} , and at τ temperature offset points, using an enumerative approach would solve for $b^2 \cdot \tau$ waveforms, involving the extensive use of lookup tables. In contrast, our approach reduces the solution to finding just six waveforms: one for the zero body bias, $V_o^Z(t)$, and one each for $\alpha_0(t)$ and $\beta_0(t)$, and three for $\sigma_p(\Delta T, t)$. The net result is a large savings in the storage and computation. Thus, the steps involved in computing the waveform at any (v_{bp}, v_{bn}) and ΔT are summarized below:

- 1) Apply equation (27) to generate the waveform $V_o^Z(t)$ at zero-body bias and zero temperature offset.
- 2) Compute and save $\alpha_0(t)$, $\beta_0(t)$ at every timestep from equations (32), (33), and (35).
- 3) Compute and save $\sigma_p(\Delta T, t)$ at every timestep from equations (34) and (38).
- 4) Use the computed $\alpha_0(t)$, $\beta_0(t)$ and $\sigma_p(\Delta T, t)$ in equation (40) to directly generate the waveform for any value of (v_{bp}, v_{bn}) and ΔT .

VI. EXPERIMENTAL RESULTS

Our results are based on standard library cells using the 45nm PTM [24], and our accuracy is measured through comparisons with the results of HSPICE [25] simulations.

A. Reduction in CSM Sensitivity Table Size

We apply our table reduction algorithm for the sensitivity parameters, $\{a_I, b_I, a_Q, b_Q\}$ and $\{c_I, r_I, c_Q, r_Q\}$ for a set of standard cells characterized using 45nm PTM [24], and demonstrate our results in Table II for a typical table, for a_I . Columns 2 through 4 show the number of entries in the reduced table using the original compression approach (Section III-A), and Columns 5 through 7 list the size of the reduced tables using our approach (Section III-B). These comparisons are shown for various bounds (2%, 5%, 10%) on the allowable error, and in each case, the optimal table size corresponds to the smallest $m \times m$ table, indexed by (V_i, V_o) , that meets the error bound. In each case, $m = 30$ for the original table size, i.e., it has 900 entries. As is seen from the table, in each case, our approach yields much smaller tables than the prior approach.

TABLE II: Results for sensitivity parameter table reduction for tables with original size = 900

Cell Type	Reduced Table Size with Error Bounds						Run Time
	Original approach			Our approach			
	2%	5%	10%	2%	5%	10%	
INV	529	484	324	225	169	100	115s
NAND2	576	484	289	196	144	81	110s
NOR2	900	784	576	324	256	169	168s
NAND3	625	529	256	169	144	81	104s
NOR3	841	729	484	289	225	144	167s
AOI21	576	529	484	196	169	100	114s
AOI22	529	484	361	225	169	81	117s

The last column of Table II shows the runtime of the algorithm for achieving reduced table sizes for the most computationally-intensive solution, where the 2% error bound must be satisfied. The runtimes are measured on a 3GHz Intel Core2Duo CPU, and correspond to the average for the $\{a_I, a_Q, b_I, b_Q\}$ and $\{c_I, c_Q, r_I, r_Q\}$ sensitivity tables, and are very reasonable, especially considering that this characterization computation must be performed only once for a given library in a given technology.

It is easy to explain why the original algorithm of Section III-A does not lead to sufficient reduction in the table size. This can primarily be related to outliers: ignoring these points causes substantial errors at these points when interpolation is used to predict the values of missing entries. On the other hand, if these are included, the large jumps at these points can result in interpolation errors at nearby points that do not correspond to outliers. These errors can only be diminished by using reduced tables of larger sizes.

B. Speedup Due to Waveform Sensitivity Models

We now present the speedup obtained using our various simplified body bias, temperature and the complete waveform sensitivity (WS) models, as proposed in Sections V-B, and V-C, respectively.

We evaluate the speedup of our models over HSPICE and over the Newton-Raphson solver (see Section IV-B) that would be used in a simple extension of existing CSMs. To calculate the above speedup, we perform our tests with each circuit example under multiple combinations of the following parameters: multiple rise/fall waveforms (1ps–100ps input ramps, in

steps of 5-10ps), various RC interconnects from the ChipA-1K, ChipB-1K and the ChipB-5K family [26] as load benchmarks reduced to π -models, multiple body bias points (169 points with $(v_{bp}, v_{bn}) \in [-0.3V, 0.3V]$, in steps of 0.05V for each parameter), and multiple temperature points (13 points with $\Delta T \in [-50^\circ C, 100^\circ C]$, in steps of 12.5 $^\circ C$).

First, we present the speedups with simplified body bias waveform sensitivity model and those of simplified temperature waveform sensitivity model independently. Then we present the speedups of the complete waveform sensitivity model. In each case, we calculate the runtimes using HSPICE, Newton-Raphson solver and our simplified waveform sensitivity models, and average these runtimes over all the test cases to arrive at final speedup results. For the test cases, we perform transient simulations and report the speedups of our algorithm over HSPICE and over the Newton-Raphson solver. Expectedly, the speedup over HSPICE is large, and is found to be about five orders of magnitude. More interestingly, our complete waveform sensitivity model achieves an average speedup of 91.81 \times , and a maximum speedup of 99.55 \times , over the Newton-Raphson solver.

1) *Body Bias Waveform Sensitivity Model:* We evaluate the speedup achieved using the standalone body bias model as presented in Section V-B. We perform evaluations at 169 body bias points within the range $(v_{bp}, v_{bn}) \in [-0.3V, 0.3V]$. All evaluations are carried at the zero temperature offset of $\Delta T = 0^\circ C$. Table III lists the speedups that are obtained by our waveform sensitivity model over HSPICE and over the Newton-Raphson Solver, for standard library cells. As can be seen from the table, the body bias waveform sensitivity model achieves an average speedup of around five orders of magnitude over HSPICE and an average speedup of 67.9 \times over the Newton-Raphson Solver.

TABLE III: Speedups obtained by the Complete Waveform Sensitivity (WS) Model over HSPICE and Newton-Raphson (NR) solver

Cell	WS Model Speedups					
	Body Bias		Temperature		Combined	
	Over HSPICE	Over NR Solver	Over HSPICE	Over NR Solver	Over HSPICE	Over NR Solver
INV	8.9e4	65.36	4.6e3	4.374	1.15e5	85.12
NAND2	9.6e4	66.29	4.9e3	4.352	1.28e5	88.15
NOR2	9.2e4	69.23	4.0e3	4.454	1.26e5	95.02
NAND3	9.6e4	66.67	4.8e3	4.313	1.29e5	89.50
NOR3	8.9e4	72.15	4.2e3	4.405	1.23e5	99.55
AOI21	10.8e4	66.80	4.8e3	4.389	1.45e5	89.72
AOI22	10.0e4	69.36	4.9e3	4.413	1.39e5	95.60

2) *Temperature Waveform Sensitivity Model:* Next, we evaluate the speedup achieved using the standalone simplified temperature waveform sensitivity model as presented in Section V-B. We perform evaluations at 13 temperature points within the range $\Delta T \in [-50^\circ C, 100^\circ C]$ with zero body biasing. Table III presents the average speedup attained over HSPICE and the Newton-Raphson solver. Compared to body bias case, these speedups are lower since we are evaluating at a much lesser number of temperature points (13 as compared to 169 in the body bias case).

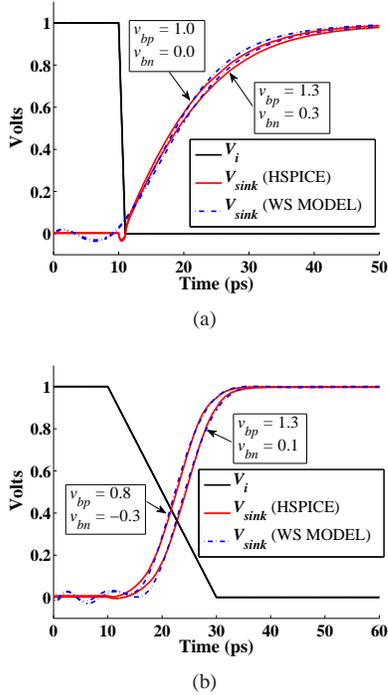


Fig. 9: The result of our simplified body bias waveform sensitivity (WS) method as compared with HSPICE, for several body bias values: (a) output waveform from an Inverter, loaded with a 20l benchmark RC interconnect, evaluated at sink node 52, and (b) output waveform from a NAND2, loaded with a 45l benchmark RC interconnect, evaluated at sink node 103.

3) *Complete Waveform Sensitivity Model*: We now present the speedup obtained with our complete body bias and temperature waveform sensitivity model as presented in Section V-C. In this case, we perform evaluations at all combinations of the 169 body bias and 13 temperature points within the range $(v_{bp}, v_{bn}) \in [-0.3V, 0.3V]$, and $\Delta T \in [-50^\circ C, 100^\circ C]$ (thus a total of 169×13 evaluations). Table III presents the average speedup attained by our complete model over HSPICE and the Newton-Raphson solver. Note that with the complete model, we are able to achieve an order of five magnitudes speedup over HSPICE. Our complete model is much faster as compared to the Newton-Raphson solver, over which we are able to achieve an average speedup of $91.81\times$, considering all temperature and body bias points.

C. Accuracy of the Waveform Sensitivity Models

In this subsection, we present the accuracy achieved by our body bias and temperature models in both waveform generation and computation of slews and delays over multiple combinations of body bias and temperature values. Through the accuracy of these waveforms and low errors in slews and delays, we also show that our assumptions of making waveform sensitivity models simplified are justified. We present accurate waveform generation both at the output node of the cell as well as the sink nodes of the RC interconnect loads, which are connected to the output node of the cell.

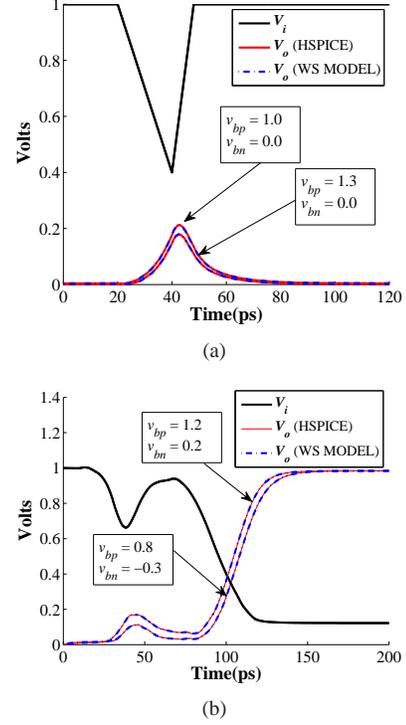


Fig. 10: Similar results of output waveform at the output node of a gate (a) for a NAND2, modeling an input glitch, and (b) for a NAND3, with a nonmonotone input.

1) *Body Bias Waveform Sensitivity Model*: The temperature offset in this part of evaluation is set to zero. Figures 9 and 10 compare representative waveforms as generated through HSPICE [25] and the simplified body bias waveform sensitivity model, when the input waveform takes any arbitrary shape either due to glitches, noise or crosstalk. We evaluate accuracies both at the output node of the cell, and the sink nodes of the interconnects which load the cell output node. Figure 9 shows the typical response of the cell at the sink nodes of the RC tree interconnect loads. The waveform is first obtained at the output node of the cell, and then evaluated at sink node using Padé-approximation of the RC interconnect circuit, and model order reduction techniques [23]. Figure 10 shows the output waveforms at the output node of the cells, with arbitrary inputs. The waveforms in some cases are coincident to the naked eye, as our algorithm yields high accuracy. This also validates the idea that α, β can be assumed to be independent of (v_{bp}, v_{bn}) , as proposed in Section V-B. Note that the initial ringing error in these waveforms is due to the use of Padé-approximation, and not due to the waveform sensitivity model.

2) *Temperature Waveform Sensitivity Model*: As with the body bias waveform sensitivity model, the simplified temperature sensitivity model as described in Section V-B yields accurate waveforms for any temperature offset value. Note that the body bias is kept at zero in all such evaluations. Figure 11 shows a set of waveforms obtained from a NOR3 cell, loaded with 33l RC interconnect network. The waveform is first obtained at the output node of the NOR3 cell, and the waveform shown is then evaluated at sink node 55. As shown in

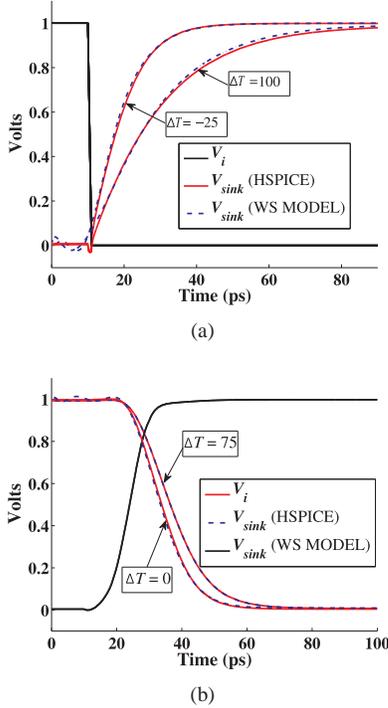


Fig. 11: The result of our simplified temperature waveform sensitivity (WS) method as compared with HSPICE, for various temperature values. Shown above are output waveforms from a NOR3, loaded with $33l$ benchmark RC interconnect, evaluated at sink node 55: (a) for a falling step input, and (b) for a slower rising input.

the figure, the temperature waveform sensitivity model yields very accurate waveforms. This also validates the simplification of $\sigma(\Delta T, t)$ values, as proposed in Section V-B.

3) *Complete Waveform Sensitivity Model*: For presenting the results in this section, we generate waveforms for multiple combinations of body bias or temperature offset values and compare the result with the corresponding waveforms obtained from HSPICE. We find that the complete model as presented in Section V-C, generates very accurate waveforms. As before, we evaluate accuracies both at the output node of the cell, and the sink nodes of the interconnects which load the cell output node. Figure 12 shows the accuracy obtained at the output node of an inverter loaded with $45l$ RC interconnect, with inputs having glitches and arbitrary shapes. Figure 13 shows the waveform evaluated at sink node 52 of $20l$ RC load interconnect for NAND2 and NOR2 cells. Our results show that a linear model for $V_o(t)$ in both body bias and temperature with simplifications as in equations (35) and (38), suffices for generation of waveforms at any combination of body bias and temperature, with sufficiently desired accuracy.

4) *Slew and Delay Errors*: We now present some more descriptive tables for the errors in delays and slews that are incurred in formulation of the complete waveform sensitivity model. For this tabulation, we work with the test cases that were mentioned at the beginning of Section VI-B, and save the delay and slew values as obtained from our complete waveform

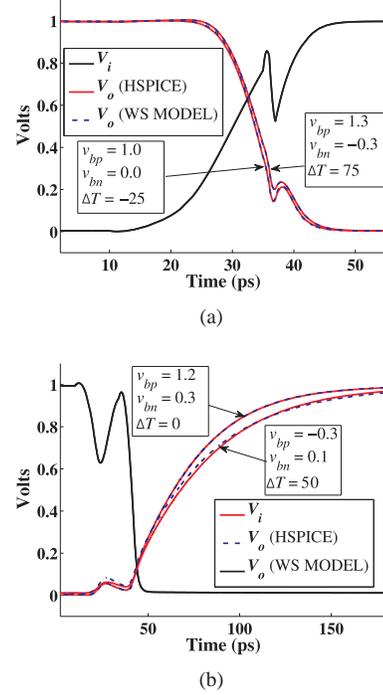


Fig. 12: The result of our complete waveform sensitivity (WS) model as compared with HSPICE, for various temperature and body bias values. Shown above are waveforms at the output node of an Inverter with (a) $45l$ as the interconnect load and an input glitch due to crosstalk, and (b) $25m$ as the interconnect load and an arbitrary input.

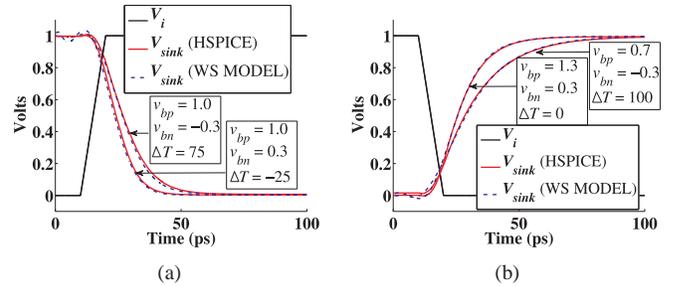


Fig. 13: Similar output waveforms from cells loaded with $20l$ benchmark RC interconnect, evaluated at farthest sink node 52: (a) the output from an NAND2 for a rising input, and (b) the output from a NOR2 for a falling input.

sensitivity model and from HSPICE. We then obtain the relative percentage error between delays and slews corresponding to the complete simplified waveform sensitivity model and of HSPICE. All such errors are tabulated. Table IV shows the mean and standard deviation of these relative errors for a NAND2 cell, over all $(v_{bp}, v_{bn}, \Delta T)$ points, presented for each combination of inputs slews and output load interconnects. It is seen that both the mean and standard deviations are small for all test cases.

A more detailed view of these rise and fall delay/slew errors is presented in Table V, for a particular test case: with a NAND2 cell loaded with $20l$ as the RC interconnect, waveforms being evaluated at sink node 52. This table shows

TABLE IV: Mean and standard deviation (St. Dev.) of the percentage errors over all ($v_{bp}, v_{bn}, \Delta T$) points, incurred by our complete waveform sensitivity model in the output rise delay and slew values for NAND2 cell, as compared to HSPICE for different input slews and output RC interconnect loads

Input Slews	RC Interconnect Loads											
	25m				33l				45l			
	Percent Delay Error		Percent Slew Error		Percent Delay Error		Percent Slew Error		Percent Delay Error		Percent Slew Error	
	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.
5ps	1.655	1.450	3.128	2.541	1.218	1.266	3.328	2.632	0.889	0.831	3.107	2.448
10ps	1.532	1.394	3.108	2.541	1.031	1.186	3.203	2.656	1.043	0.960	2.407	2.304
20ps	1.328	1.376	3.095	2.552	0.883	0.950	2.832	2.596	1.557	1.338	2.045	2.156
50ps	1.030	1.164	2.826	2.568	1.397	1.090	2.353	2.530	4.026	3.992	1.876	2.167
100ps	1.244	1.192	1.183	1.405	2.245	1.727	2.087	2.331	1.785	1.813	1.869	2.347

TABLE V: Percent delay and slew errors for a NAND2 cell at various temperature offsets, over all v_{bp}, v_{bn} points

ΔT (°C)	Percent Delay Errors			Percent Slew Errors		
	Max.	Min.	Mean	Max.	Min.	Mean
-50.0	1.43	0.06	0.47	4.63	0.09	1.75
-37.5	1.15	0.02	0.39	4.33	0.19	1.78
-25.0	1.23	0.03	0.40	3.71	0.16	1.64
-12.5	1.14	0.01	0.49	2.02	0.10	0.82
0.0	1.29	0.08	0.44	1.23	0.08	0.70
12.5	1.20	0.09	0.40	1.51	0.06	0.78
25.0	1.15	0.05	0.45	2.50	0.13	1.04
37.5	1.34	0.09	0.66	4.16	0.04	1.53
50.0	2.18	0.38	1.11	4.81	0.06	2.65
62.5	1.96	0.22	0.81	5.48	0.27	2.79
75.0	1.27	0.02	0.39	5.94	0.26	2.92
87.5	1.18	0.06	0.68	6.87	0.40	3.19
100.0	2.16	0.63	1.41	8.93	0.35	3.74

the distribution of the delay/slew errors over all (v_{bp}, v_{bn}) points, but at different ΔT values. Column 1 of the table lists the temperature offsets from the nominal temperature of 25°C at which the waveforms are evaluated. Columns 2 to 4 present, for the value of ΔT listed in column 1, the maximum, the minimum and the mean of the percentage delay errors obtained over all v_{bp}, v_{bn} points. Similarly, columns 5 to 7 present the maximum, the minimum and the mean percentage slew errors obtained over all v_{bp}, v_{bn} points at that temperature offset. Clearly, at all temperature offsets, both the mean delay and slew errors over all v_{bp}, v_{bn} points are contained within 4%.

We thus observe from Tables IV and V that only a small error is incurred in both delays and slews over all combinations of v_{bp}, v_{bn} , and ΔT points, validating the use of our waveform sensitivity model in predicting the delay and slews over the entire range of body bias and temperature. Very similar observations were made for other standard cells in the library.

VII. CONCLUSION

A simple extension of existing CSMs to incorporate the effects of body bias and temperature in the CSM framework results in excessive increase in library memory and solver runtime. We present a novel approach to incorporate body bias and temperature effects into current source models. We develop sensitivity model for capturing variations in CSM components with body bias and temperature, with compaction of the resulting tables of these model parameters. We incorporate this sensitivity model into the mainstream CSM solver framework, and develop a new model for capturing waveform sensitivity with body bias and temperature, which allows us to

compute waveforms at multiple combinations of body bias and temperature points with massive savings in computation. The results demonstrate the effectiveness of our compaction scheme and the waveform sensitivity model in achieving HSPICE level accuracy with high speedups both over HSPICE and conventional CSM solvers.

REFERENCES

- [1] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9-V, 150 MHz, 10-mW, 4 mm², 2-D discrete cosine transform core processor with variable threshold-voltage (VT) scheme," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 166–167, 1996.
- [2] J. W. Tschanz, J. Kao, S. G. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 1396–1402, November 2002.
- [3] J. Tschanz, N. Kim, S. Dighe, J. Howard, G. Ruhl, S. Vanga, S. Narendra, Y. Hoskote, H. Wilson, C. Lam, M. Shuman, C. Tokunaga, D. Somasekhar, S. Tang, D. Finan, T. Karnik, N. Borkar, N. Kurd, and V. De, "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 292–604, 2007.
- [4] S. S. Sapatnekar, *Timing*. Boston, MA: Springer, 2004.
- [5] F. Dartu, N. Menezes, and L. Pileggi, "Performance computation for precharacterized CMOS gates with RC loads," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 15, pp. 544–553, May 1996.
- [6] J. F. Croix and D. F. Wong, "Blade and Razor: Cell and interconnect delay analysis using current-based models," in *Proceedings of the Design Automation Conference*, pp. 386–389, 2003.
- [7] I. Keller, K. Tseng, and N. Verghese, "A robust cell-level crosstalk delay change analysis," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 147–154, 2004.
- [8] P. Li and E. Acar, "A waveform independent gate model for accurate timing analysis," in *Proceedings of the International Conference on Computer Design*, pp. 363–365, 2005.
- [9] C. Amin, C. Kashyap, N. Menezes, K. Killpack, and E. Chiprout, "A multi-port current source model for multiple-input switching effects in CMOS library cells," in *Proceedings of the Design Automation Conference*, pp. 247–252, 2006.
- [10] C. Kashyap, C. Amin, N. Menezes, and E. Chiprout, "A nonlinear cell macromodel for digital applications," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 678–685, 2007.
- [11] N. Menezes, C. Kashyap, and C. Amin, "A "true" electrical cell model for timing, noise, and power grid verification," in *Proceedings of the Design Automation Conference*, pp. 462–467, 2008.
- [12] B. Amelifard, S. Hatami, H. Fatemi, and M. Pedram, "A current source model for CMOS logic cells considering multiple input switching and stack effect," in *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 568–573, 2008.
- [13] A. Goel and S. Vrudhula, "Current source based standard cell model for accurate signal integrity and timing analysis," in *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 574–579, 2008.

- [14] S. Raja, F. Varadi, M. Becer, and J. Geada, "Transistor level gate modeling for accurate and fast timing, noise, and power analysis," in *Proceedings of the Design Automation Conference*, pp. 456–461, 2008.
- [15] V. Gerousis, "Design and modeling challenges for 90 nm and 50 nm," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 353–360, 2003.
- [16] G. Ono, M. Miyazaki, M. Tanaka, N. Ohkubo, and T. Kuwahara, "Temperature referenced supply voltage and forward-body-bias control (TSFC) architecture for minimum power consumption," in *Proceedings of the European Solid State Circuits Conference*, pp. 391–394, 2004.
- [17] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "Body bias voltage computations for process and temperature compensation," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 16, pp. 249–262, March 2008.
- [18] Y. Zhan, S. V. Kumar, and S. S. Sapatnekar, "Thermally aware design," *Foundations and Trends in Electronic Design Automation*, vol. 2, no. 3, pp. 255–370, 2008.
- [19] K. Bowman, J. Tschanz, C. Wilkerson, S. Lu, T. Karnik, V. De, and S. Borkar, "Circuit techniques for dynamic variation tolerance," in *Proceedings of the Design Automation Conference*, pp. 4–7, 2009.
- [20] J. Tschanz, S. Narendra, A. Keshavarzi, and V. De, "Adaptive circuit techniques to minimize variation impacts on microprocessor performance and power," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 9–12, 2005.
- [21] Y. Taur and T. H. Ning, *Fundamentals of modern VLSI devices*. Cambridge, UK: Cambridge University Press, Second Edition, 2009.
- [22] J. F. Croix and D. F. Wong, "A fast and accurate technique to optimize characterization tables for logic synthesis," in *Proceedings of the Design Automation Conference*, pp. 337–340, 1997.
- [23] P. R. O'Brien and T. L. Savarino, "Modeling the driving point characteristic of resistive interconnect for accurate delay estimation," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 512–515, 1989.
- [24] Predictive Technology Model. <http://www.eas.asu.edu/~ptm>.
- [25] <http://www.synopsys.com/products/mixedsignal/hspice/hspice.html>.
- [26] Z. Li, C. N. Sze, C. J. Alpert, J. Hu, and W. Shi, "Making fast buffer insertion even faster via approximation techniques," in *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 13–18, 2005.
- [27] G. A. Baker and P. G. Morris, *Padé Approximants*. Cambridge, MA: Cambridge University Press, Second Edition, 1996.

APPENDIX

Proof of Theorem 1

At each time step, combining the nonlinear equation (23) with (29), and A, B given by (24), (25), we can write $\alpha(t)$ as:

$$\alpha(t) = \frac{BC_1}{A} \frac{\partial V_o^n}{\partial v_{bp}} + \frac{hC_2}{A} \frac{\partial V_{C_2}^n}{\partial v_{bp}} + \frac{B}{A} \left(h \frac{\partial I_p}{\partial v_{bp}} + \frac{\partial Q_p}{\partial v_{bp}} - \frac{\partial Q_p^n}{\partial v_{bp}} \right) \quad (41)$$

Writing I_p, Q_p as linear functions of v_{bp}, v_{bn} from (4), (5), we get

$$\begin{aligned} \alpha(t) = & \frac{BC_1}{A} \frac{\partial V_o^n}{\partial v_{bp}} + \frac{hC_2}{A} \frac{\partial V_{C_2}^n}{\partial v_{bp}} + \frac{B}{A} \left[h \frac{\partial I_p^Z}{\partial V_o} \frac{\partial V_o}{\partial v_{bp}} L_I(v_{bp}, v_{bn}) + \right. \\ & \frac{\partial Q_p^Z}{\partial V_o} \frac{\partial V_o}{\partial v_{bp}} L_Q(v_{bp}, v_{bn}) - \frac{\partial Q_p^{Z,n}}{\partial V_o^n} \frac{\partial V_o^n}{\partial v_{bp}} L_Q^n(v_{bp}, v_{bn}) + \\ & h I_p^Z \left(a_I + \frac{\partial a_I}{\partial V_o} \frac{\partial V_o}{\partial v_{bp}} v_{bp} + \frac{\partial b_I}{\partial V_o} \frac{\partial V_o}{\partial v_{bp}} v_{bn} \right) + \\ & Q_p^Z \left(a_Q + \frac{\partial a_Q}{\partial V_o} \frac{\partial V_o}{\partial v_{bp}} v_{bp} + \frac{\partial b_Q}{\partial V_o} \frac{\partial V_o}{\partial v_{bp}} v_{bn} \right) - \\ & \left. Q_p^{Z,n} \left(a_Q^n + \frac{\partial a_Q^n}{\partial V_o^n} \frac{\partial V_o^n}{\partial v_{bp}} v_{bp} + \frac{\partial b_Q^n}{\partial V_o^n} \frac{\partial V_o^n}{\partial v_{bp}} v_{bn} \right) \right] \end{aligned}$$

where L_I and L_Q are defined as in equations (6) and (7), respectively, and L_Q^n corresponds to the evaluation of L_Q at time step n . Further, $\partial V_{C_2} / \partial v_{bp}$ can be calculated using equation (26).

Recognizing that $\frac{\partial V_o}{\partial v_{bp}} = \alpha$, and at time step n , $\alpha^n = \frac{\partial V_o^n}{\partial v_{bp}}$, collecting all terms multiplied by α , the result of equation (32) follows immediately. The derivation of equation (33) is analogous. Note

that $\{a_I, a_Q, b_I, b_Q\}$ are independent of body bias, being functions of (V_i, V_o) only, but appear as functions of (v_{bp}, v_{bn}) since V_o dynamically changes with body bias during simulation. ■

Proof of Theorem 2

The derivation of σ follows along the same lines as for α and β . At each time step, combining the nonlinear equation (23) with (31), and A, B given by (24), (25), we can write $\sigma(t)$ as:

$$\sigma(t) = \frac{BC_1}{A} \frac{\partial V_o^n}{\partial \Delta T} + \frac{hC_2}{A} \frac{\partial V_{C_2}^n}{\partial \Delta T} + \frac{B}{A} \left(h \frac{\partial I_p}{\partial \Delta T} + \frac{\partial Q_p}{\partial \Delta T} - \frac{\partial Q_p^n}{\partial \Delta T} \right) \quad (42)$$

Writing I_p, Q_p as second order functions of ΔT from (10), (11), we get

$$\begin{aligned} \sigma(t) = & \frac{BC_1}{A} \frac{\partial V_o^n}{\partial \Delta T} + \frac{hC_2}{A} \frac{\partial V_{C_2}^n}{\partial \Delta T} + \frac{B}{A} \left[h \frac{\partial I_p^Z}{\partial V_o} \frac{\partial V_o}{\partial \Delta T} S_I(\Delta T) + \right. \\ & \frac{\partial Q_p^Z}{\partial V_o} \frac{\partial V_o}{\partial \Delta T} S_Q(\Delta T) - \frac{\partial Q_p^{Z,n}}{\partial V_o^n} \frac{\partial V_o^n}{\partial \Delta T} S_Q^n(\Delta T) + \\ & h I_p^Z \left(c_I + 2r_I \Delta T + \frac{\partial c_I}{\partial V_o} \frac{\partial V_o}{\partial \Delta T} \Delta T + \frac{\partial r_I}{\partial V_o} \frac{\partial V_o}{\partial \Delta T} \Delta T^2 \right) + \\ & Q_p^Z \left(c_Q + 2r_Q \Delta T + \frac{\partial c_Q}{\partial V_o} \frac{\partial V_o}{\partial \Delta T} \Delta T + \frac{\partial r_Q}{\partial V_o} \frac{\partial V_o}{\partial \Delta T} \Delta T^2 \right) - \\ & \left. Q_p^{Z,n} \left(c_Q^n + 2r_Q^n \Delta T + \frac{\partial c_Q^n}{\partial V_o^n} \frac{\partial V_o^n}{\partial \Delta T} \Delta T + \frac{\partial r_Q^n}{\partial V_o^n} \frac{\partial V_o^n}{\partial \Delta T} \Delta T^2 \right) \right] \end{aligned}$$

where S_I and S_Q are defined as in equations (12) and (13), respectively, and S_Q^n corresponds to the evaluation of S_Q at time step n . Further, $\partial V_{C_2} / \partial \Delta T$ can be calculated using equation (26).

Since $\frac{\partial V_o}{\partial \Delta T} = \sigma$, and at time step n , $\sigma^n = \frac{\partial V_o^n}{\partial \Delta T}$, collecting all terms multiplied by σ on left hand side, the result of equation (34) follows immediately. Note that $\{c_I, c_Q, r_I, r_Q\}$ are again independent of temperature, being functions of (V_i, V_o) only. ■



Saket Gupta Saket Gupta received the B.Tech. degree in electronics and computer engineering from Indian Institute of Technology, Roorkee, India, in 2008, and the M.S. degree in electrical engineering from the University of Minnesota, Minneapolis, in 2011. He is currently pursuing his PhD degree in electrical engineering from the University of Minnesota, Minneapolis. His research interests include development and implementations of CAD-related timing tools, and high-performance circuits and architectures that are variations and aging resilient.

His other interests include numerical modeling and simulation, and parallel programming on multicores and GPUs.



Sachin S. Sapatnekar Sachin S. Sapatnekar (F03) received the Ph.D. degree from the University of Illinois at Urbana-Champaign in 1992. He is currently with the University of Minnesota, where he holds the Distinguished McKnight University Professorship and the Robert and Marjorie Henle Chair in Electrical and Computer Engineering. His research interests are in the analysis and optimization of nanometer-scale VLSI circuits.

He has held positions on the editorial boards of various journals and is currently the Editor-in-Chief of the Transactions on CAD. He has served as Technical Program Chair and General Chair for the Design Automation Conference, the International Symposium on Physical Design, and the Tau workshop. He is a recipient of the NSF Career Award, six conference Best Paper Awards, and the SRC Technical Excellence Award.