

A Sub-Threshold Noise Transient Simulator Based on Integrated Random Telegraph and Thermal Noise Modeling

Marco Donato, *Member, IEEE*, R. Iris Bahar, *Senior Member, IEEE*, William R. Patterson, *Life Member, IEEE*, and Alexander Zaslavsky

Abstract—Near-threshold and sub-threshold voltage designs have been identified as possible solutions to overcome the limitations introduced by energy consumption in modern very large scale integration circuits. However, as we approach sub-10 nm transistor technology, aggressive voltage, and gate length scaling will reduce the reliability of logic circuits due to the increasing impact of noise and variability effects. Therefore, designers need new tools to simulate logic circuits in the presence of noise. Time-domain analysis helps understand how transient faults affect a circuit and can guide designers in producing noise-resistant circuitry. However, standard approaches to modeling intrinsic noise sources in the time domain are computationally expensive. Moreover, small noise-driven fluctuations in electron occupation of circuit nodes introduce time-varying biasing point fluctuations, increasing the modeling complexity. To address these challenges, this paper introduces a new approach to modeling thermal noise and random telegraph signal noise directly in the time domain by developing and solving a series of stochastic differential equations. In comparisons to traditional SPICE-based simulations, our approach can provide three orders of magnitude speedup in simulation time without sacrificing accuracy. Moreover, we introduce a novel, iterative threshold-crossing algorithm, aimed at the efficient sampling of rare noise transients. We show that Monte-Carlo simulations based on this approach can detect rare high-amplitude single event transients that would be impossible to uncover with standard transient simulators.

Index Terms—Random telegraph signal (RTS) noise, reliability, simulation, single-event transients, thermal noise.

I. INTRODUCTION

A MAJOR advantage of CMOS digital circuits is their relatively high signal-to-noise ratio, making them resilient to various sources of error. Even so, their error resiliency is being challenged by the increasing impact of intrinsic noise and variability sources. Much of the research on error rate

Manuscript received October 21, 2016; revised March 14, 2017; accepted May 16, 2017. Date of publication June 20, 2017; date of current version February 16, 2018. This work was supported by NSF under Grant CCF-1525486. This paper was recommended by Associate Editor A. Srivastava. (*Corresponding author: Marco Donato.*)

M. Donato was with the School of Engineering, Brown University, Providence, RI 02912 USA. He is now with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: marco_donato@alumni.brown.edu).

R. I. Bahar, W. R. Patterson, and A. Zaslavsky are with the School of Engineering, Brown University, Providence, RI 02912 USA.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2017.2717705

analysis and fault tolerance techniques has been targeted at radiation-induced single-event upsets, specifically, in SRAMs and latches [1]–[4], and more recently in combinational logic circuits [5]–[11]. According to the International Technology Roadmap for Semiconductors (ITRS), we can expect transistors to have gate length $L_g = 5$ nm and $V_{DD} < 0.6$ V by 2028 [12]. Even more aggressive voltage scaling has been suggested as near- and subthreshold operation design has gained popularity [13]–[15]. As the gate length of the transistors enters the sub-10 nm range and the operating voltage is scaled to a fraction of a volt, the number of electrons responsible for the total charge of a CMOS node is greatly reduced. For a minimum sized inverter with $L_g = 16$ nm and $V_{DD} = 0.9$ V, we can expect the number of electrons $N \sim 900$. However, for the 5 nm technology predicted by ITRS for 2028, the number of electrons is expected to go down to $N < 100$. As a result, combinational logic circuits will become more and more susceptible to intrinsic noise sources that would otherwise be a concern only for specific design domains.

For example, thermal noise, which is the current or voltage fluctuation on a conductive component associated with the electron's thermal agitation, is usually addressed as an issue only in analog circuits. To a first approximation, a transistor in a CMOS circuit can be represented as an RC equivalent circuit. The standard deviation of thermal noise is equal to $\sqrt{kT/C}$, where C is equivalent to the total capacitance attached to the CMOS gate output node. Since C decreases with technology scaling, thermal fluctuations will correspondingly increase. If we consider again the two technology processes above, as transistor dimensions scale down from 16 to 5 nm, we expect thermal noise voltage fluctuations to increase from a few mV to tens of mV (i.e., an increase by an order of magnitude). Hence, while transistors manufactured in current technology processes are still not affected by thermal noise, this has been identified as one of the factors that will lead to the end of technology scaling as we know it [16]. Moreover, random telegraph signal (RTS) noise, a phenomenon related to the trapping and de-trapping of carriers in the gate oxide, has become a growing concern for digital circuit designers. While scaling the transistor's dimensions down reduces the number of traps per device, each trap has a greater impact on the threshold voltage shift. Therefore, these trapping events produce discrete step changes in the drain current which can be as large as 80% of the total current [17].

Time-domain noise analysis offers a detailed description of the dynamic response of a circuit in the presence of noise and may be used as a guide in designing noise-immune circuitry. Several works have also targeted the modeling of RTS noise, e.g., [18]–[20]. In particular, the work of Mahmutoglu and Demir [18] and Wirth *et al.* [19] have shown the importance of taking into account time-varying biasing conditions for the correct estimation of RTS noise characteristics. The same consideration applies to thermal noise modeling. However, state-of-the-art time-domain thermal noise simulators are SPICE-based simulators, which are computationally very costly, especially, if the goal is to observe rare fault-inducing noise transients.

Accurate modeling of thermal noise in sub-threshold circuits requires a careful choice of the proper probability distribution to match the physical behavior of these circuits. Thermal noise has commonly been modeled using additive white Gaussian noise, where the voltage fluctuations follow a Gaussian distribution with zero mean and fixed standard deviation. However, when the total charge at the output of a CMOS gate is very low, even fluctuations of a few electrons can bring substantial changes to the operating point of the transistors, implying a nonlinear behavior that requires adopting time varying parameters to compute the noise. Moreover, it has been shown that thermal fluctuations in circuits operating in the sub-threshold regime follow a Poissonian rather than Gaussian distribution [21].

In order to address these issues, we recently proposed a fast simulation framework for the analysis of noise transients induced by thermal noise fluctuations [22]. In that work, we modeled CMOS gates in the presence of noise using stochastic differential equations (SDEs). The resulting simulator allowed us to accurately simulate thermally induced noise transients with a three orders of magnitude speedup compared to traditional SPICE-based simulators. In addition, we introduced a novel iterative algorithm that allows extracting rare transient events that would not be possible to simulate otherwise.

In this paper, we expand on and complete our previous framework [22] in several significant ways to improve accuracy and demonstrate broader applicability.

- 1) We improve the accuracy of the simulation framework by replacing the single iterative functions presented in [22] with a matrix representation of the SDE system.
- 2) We present a more detailed definition of the iterative threshold-crossing algorithm introduced in [22] with the addition of an empirical method for obtaining the circuit's time-to-failure.
- 3) We incorporate RTS noise in the simulation framework by integrating the effect of carrier trapping and detrapping events on the Poisson charging rates of circuit nodes.
- 4) Based on these improvements we present a comprehensive analysis of the impact of intrinsic noise sources on the circuits' reliability.

The remainder of this paper is organized as follows. Section II provides an overview of related work. In Section III,

we present the derivation of the SDEs used for circuit time-domain modeling and introduce the conjoint model for thermal and RTS noise. In Section IV, we describe the implementation of our simulator as well as the methodology used for fast detection of thermally induced SETs from [22]. In addition, we highlight the algorithm modification needed to include RTS noise modeling. We report simulation results in Section V and provide concluding remarks in Section VI.

II. BACKGROUND AND RELATED WORK

A. Thermal Noise Modeling

In recent years, there have been several efforts to address the design of thermal-noise-tolerant circuit architectures [23]–[26]. All these works modeled the thermal noise by scaling the noise standard deviation up to a value that would allow an SET to be observed more easily in the logic circuits. While this approach does allow for some means of evaluating the relative noise robustness of different circuit designs, the noise itself is ill-defined. In fact, while increasing the noise standard deviation to an unrealistically high value, such as 60 mV RMS, makes it possible to rapidly generate high-amplitude fault-inducing events, these events do not reflect the transient behavior of real signals. Instead, the goal of the work described in this paper is to develop a simulation framework that allows for fast detection of physically accurate thermal noise transients. More specifically, our approach is to develop a stochastic representation of noise fluctuations for a device operated in the sub-threshold region.

Note that thermal noise for transistors in sub-threshold operation follows statistics that are fundamentally different from those used in classical above-threshold circuits. It has been shown that for transistors in sub-threshold operation, electron rate fluctuations can be described using a two-sided Poisson process whose rates can be derived from the forward and reverse components of the transistors' drain currents [21]. For transistors biased in sub-threshold, the drain current is given by

$$I_D = I_0 \exp\left(\frac{qV_{gs}}{mkT}\right) \exp\left(\frac{qV_{ds}\lambda_D}{kT}\right) \left[1 - \exp\left(-\frac{qV_{ds}}{kT}\right)\right] \quad (1)$$

where λ_D is the DIBL parameter [27] and m and I_0 are technology-dependent factors. The drain current in (1) represents a deterministic quantity: the average net drain current produced by electrons traveling from source to drain and from drain to source. The two opposing forward and reverse electron currents, I_f and I_r , are random in nature, and can be treated as two independent Poisson processes [21]. The rates for these two Poisson processes can be obtained by simply dividing the average currents, I_f and I_r , by the electron charge q . For an nMOS transistor we can derive the rates expressed in terms of number of electrons per unit time as

$$\mu_n = \frac{I_f}{q} = \frac{I_0}{q} \exp\left(\frac{qV_{ds}\lambda_D}{kT}\right) \exp\left(\frac{qV_{gs}}{mkT}\right) \quad (2)$$

$$\lambda_n = \frac{I_r}{q} = \mu_n \exp\left(\frac{-qV_{ds}}{kT}\right). \quad (3)$$

Accordingly, we can define the two stochastic processes describing the forward and reverse electron flows as

$$\begin{aligned} X_f &\sim \text{Pois}(\mu_n) \\ X_r &\sim \text{Pois}(\lambda_n) \end{aligned} \quad (4)$$

where X_f and X_r are time series of Poisson-distributed random variables. In order to apply these definitions to the circuit's transient behavior, let us consider the example of an nMOS transistor discharging a capacitor C . We can write the current I_D discharging the capacitor as

$$\frac{dV_C}{dt}C = -I_D. \quad (5)$$

Following stochastic calculus conventions [28], and using the definitions from (4), we can write the stochastic counterpart of (5) as:

$$d\hat{V}_C = -\frac{q}{C}(dX_f - dX_r) = -\frac{q}{C}dX \quad (6)$$

where $q dX = I_D dt$. Integrating the resulting equation over a time interval $[0, t]$ we obtain the following solution of the SDE:

$$\hat{V}_C(t) = V_C(0) - \int_0^t \frac{q}{C} dX. \quad (7)$$

In this solution, $V_C(0)$ represents the initial voltage on the capacitor and the voltage transient output is entirely constructed from the net count of the electrons flowing in the channel via a random process. This definition sets our approach apart from previous thermal noise models, in which the noise was considered as an additive feature to deterministic voltage and current variables. It is important to recognize that the equilibrium net current can be retrieved from the expected value of the net charge

$$\begin{aligned} I_D &= q\mathbf{E}[X] = q(\mathbf{E}[X_f] - \mathbf{E}[X_r]) \\ &= q(\mu_n - \lambda_n) = I_f - I_r \end{aligned} \quad (8)$$

in which we have used the equivalence between the expected value of the two Poisson processes and the forward and reverse current components.

The extension of this model to the inverter gate can be easily constructed by considering that the sum of two or more independent Poisson processes is still a Poisson process with a rate given by the sum of the original processes [29].

As an example, consider the inverter in Fig. 1. Here, we have adopted the convention of labeling the rates associated with the current components flowing into the output node, i.e., charging the output capacitance, as λ and the rates for the opposing discharging process as μ . The rates for the charging/discharging processes can be summed together and the original four processes can be reduced to two cumulative opposing Poisson processes with rates $\lambda_{\text{inv}} = \lambda_n + \lambda_p$ and $\mu_{\text{inv}} = \mu_n + \mu_p$. In our formulation, we extend this simplification to any CMOS gate, regardless of the complexity of the circuit, by combining the rates of all the transistors connected to the output node. The two-sided Poisson model has been used to create a probabilistic framework with the goal of estimating the failure rate of memory storage elements due

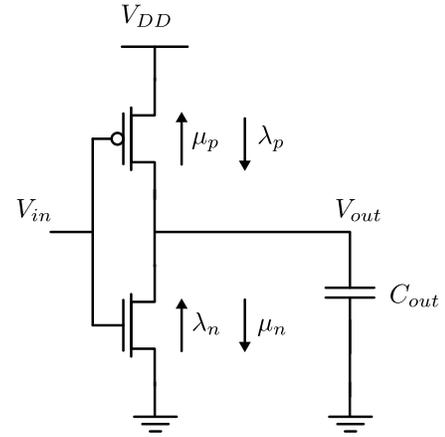


Fig. 1. Inverter rates for nMOS and pMOS transistors.

to thermal noise [30]. Specifically, the framework used projected data from the ITRS to model a cross-coupled inverter in a 10 nm technology node and was used to investigate the failure-in-time due to thermal noise as a function of other parameters such as fabrication process, V_{DD} , and temperature variations. This paper differs from [30] in two ways; we aim to describe the dynamic response of the circuits to voltage transients generated by thermal noise and, while doing so, we extend the study to more complex circuit architectures with multiple logic gates.

B. RTS Noise Modeling

RTS noise is another intrinsic noise source that is observable in modern submicron CMOS circuits. It is manifested as discrete steps in the transistor's drain current due to carrier trapping and detrapping in the gate oxide. An important result from [31] is that $1/f$ noise and RTS noise are linked to the same physical phenomenon. In fact, the $1/f$ spectrum can be derived by combining Lorentzian components that have a $1/f^2$ shape and are associated with individual traps. For this reason, larger devices, which are likely to have more traps in the oxide, are affected by $1/f$ noise, while nanoscale devices exhibit quantized noise current fluctuations.

One of the first attempts to create an RTS model to be used for SPICE transient simulations was presented in [20] where the authors used a Gaussian process filtered by a 2-stage circuit that consisted of an RC low-pass filter followed by a comparator. The effectiveness of the model was shown by simulating SRAM cells and ring oscillators. While correctly reproducing a $1/f^2$ -shaped spectrum, the approach had several limitations. First, it only allowed for simulating single traps with arbitrary values for the trapping and detrapping rates, instead of considering the statistical trap distribution. Moreover, the probability of a trap capturing or emitting carriers is a function of the biasing conditions of the circuit. More accurate models that take into account the nonstationary nature of RTS noise are reported in [19] and [32]–[34], on which we base the RTS statistical modeling used in our simulator. Other works [35], have specifically targeted the modeling of oxide traps in HfO_2 and their effect on the variability of FinFET devices, and could be used to further improve the model accuracy.

In modeling RTS noise, we combine a static trap characterization for each transistor with the dynamic characterization of trap occupation. The static trap characterization defines the number of traps per device, the depth of each trap in the gate oxide, and the initial trap state. While this configuration varies from device to device, we assume that both the number of traps and their position do not change over time. The number of traps can be sampled from a Poisson distribution whose average is related to the trap density per unit area and energy, N_T , while the trap position is uniformly distributed across the whole oxide thickness [32].

RTS noise transient behavior can be described by three main parameters associated with each active trap, namely the current noise fluctuation relative to the mean current, $\Delta I/I$, and the capture and emission time constants τ_c and τ_e . In particular, given an initial state, which could be either filled or empty, the trap will either capture or emit an electron, and the time between these events follows an exponential distribution. Once a trap captures an electron, it will produce a discrete step ΔI in the drain current. The current noise relative amplitude for each filled trap depends on technology parameters and the depth of the trap in the oxide d_{tr}

$$\frac{\Delta I}{I} = \frac{q}{mV_t WLC_{ox}} \left(1 - \frac{d_{tr}}{T_{ox}}\right) \quad (9)$$

where m is the transistor ideality factor and V_t is the thermal voltage. Notice that for a trap at the silicon-to-oxide interface, the relative noise will be at its maximum value, and it becomes smaller as the trap's location moves deeper into the oxide. The values of the average trapping and detrapping event times can be expressed using the following relationships [32]:

$$\beta = \frac{\tau_c}{\tau_e} = g \exp\left(\frac{E_T - E_F}{kT}\right) \quad (10)$$

$$\tau_c = \tau \cdot (\beta + 1) \quad (11)$$

$$\tau_e = \frac{\tau_c}{\beta}. \quad (12)$$

The total time constant τ for a given trap is calculated as [36]

$$\tau = \tau_0 e^{\gamma d_{tr}} \quad (13)$$

$$\tau_0 = 1/(n_0 \cdot v_t \cdot \sigma) \quad (14)$$

$$\gamma = \frac{\sqrt{2m^* \Phi_{barrier}}}{\hbar} \quad (15)$$

where γ and τ_0 are technological parameters depending on the characteristic of the oxide, n_0 is the carrier density, v_t is the thermal velocity, and σ is the capture cross section. Equation (10) shows the exponential dependency of the capture to emission time ratio on the trap energy level relative to the Fermi level $E_T - E_F$. This dependency also highlights the nonstationary nature of the RTS noise events as the difference between the energy levels depends on the transistor's biasing conditions [32]. The technology parameters used to implement the model have been extracted from the SPICE BSIMCMG model for a 7 nm FinFET process [37], [38] and from various reports from [39]–[43]. A list of these parameters for SiO₂ and HfO₂ oxides is shown in Table I. Fig. 2 shows the values

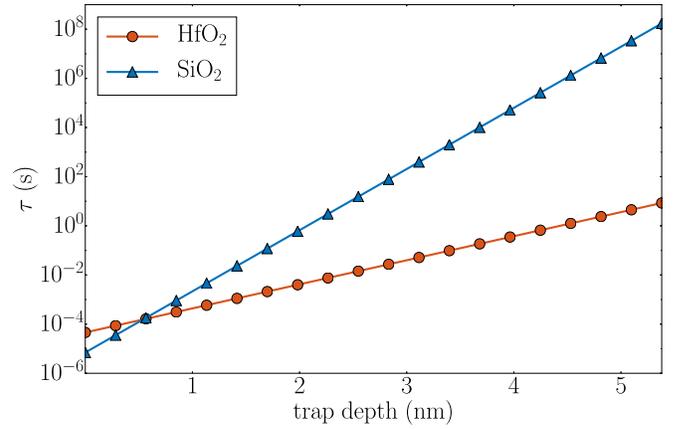


Fig. 2. Trap time constant as a function of the depth in the oxide for SiO₂ and HfO₂ with $T_{ox} = 5.38$ nm.

TABLE I
TECHNOLOGY PARAMETERS USED FOR DERIVING
THE RTS NOISE RATES AND AMPLITUDES

N_T	$3 \times 10^{18} \text{ cm}^{-3} \text{ eV}^{-1}$	
m_e^*	SiO ₂	HfO ₂
	$0.4 \cdot m_0$	$0.17 \cdot m_0$
$\phi_{barrier}$	3.1 eV	1.13 eV
τ_0	$7 \times 10^{-6} \text{ s}$	$4.56 \times 10^{-5} \text{ s}$
γ	$5.72 \times 10^7 \text{ cm}^{-1}$	$2.25 \times 10^7 \text{ cm}^{-1}$

of trap time constants as a function of the trap depth in the oxide.

In the following section, we present the derivation of the nodal SDEs used by our simulator for computing the output voltage of CMOS logic gates in the presence of noise.

III. NOISE-DRIVEN SIMULATION

Consider the circuit in Fig. 3(a) with transient response of the voltage at node V_{outA} shown in Fig. 3(b). The overshoot shown at the beginning of the transition is caused by the input/output coupling through the Miller capacitance, C_{MA} , while the tail toward the end of the transition is due to feedback coupling through the Miller capacitance of the second stage C_{MB} . Note that this “heavy tail” is noticeable in this circuit when operated in sub-threshold, due to increased propagation delay; it would be negligible if the same circuit were operated at standard above-threshold voltages.

These two effects can be modeled by considering Kirchoff's current law at the output node of the first inverter. The equilibrium state at V_{outA} when the node charge is not subject to any noise can be expressed as

$$\begin{aligned} \frac{dV_{outA}}{dt} C_L = & \frac{d(V_{in} - V_{outA})}{dt} C_{MA} + I_P - I_N \\ & + \frac{d(V_{outB} - V_{outA})}{dt} C_{MB}. \end{aligned}$$

As previously mentioned in Section II, we define the sub-threshold current as a result of two competing stochastic

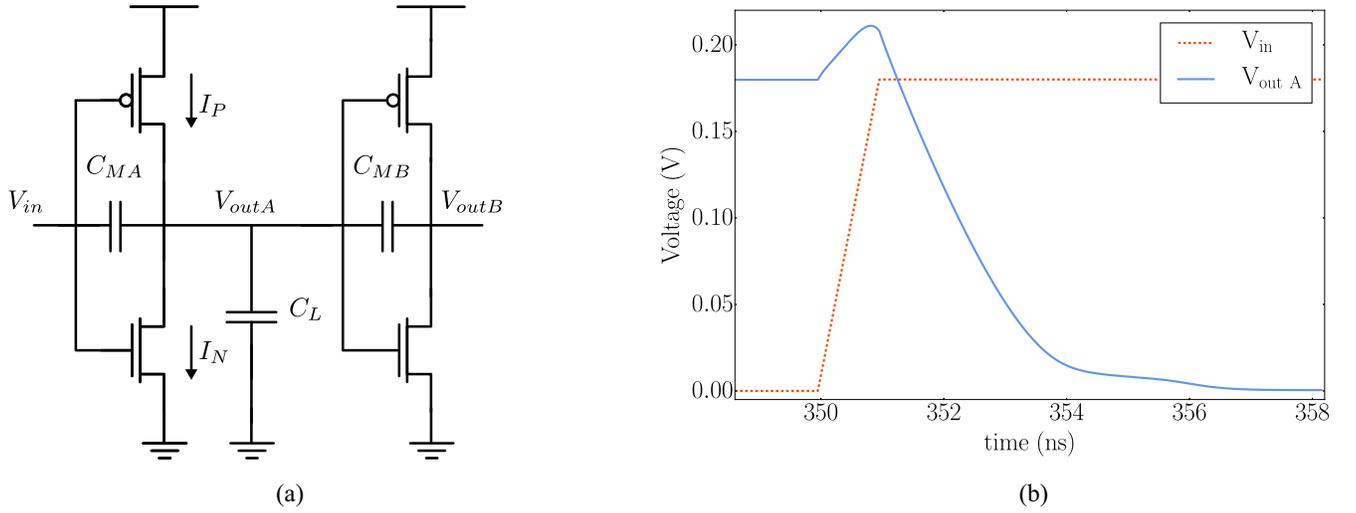


Fig. 3. (a) Two inverters connected in series and (b) the transient response at node V_{outA} to an input step function at V_{in} .

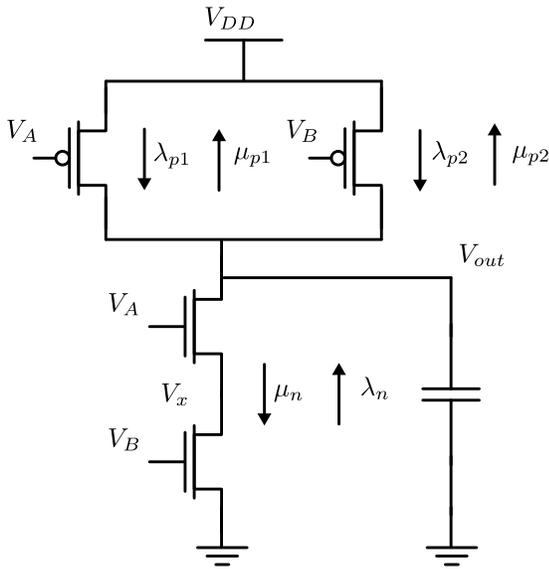


Fig. 4. Poisson rates for a NAND gate, as shown in [44].

Poisson processes as:

$$\hat{I}_P - \hat{I}_N = q(X_{ch} - X_{dis}) = \hat{I} \quad (16)$$

where $X_{ch} \sim \text{Pois}(\lambda_p + \lambda_n)$ and $X_{dis} \sim \text{Pois}(\mu_p + \mu_n)$.

Following the same approach used for solving (6), and grouping together the terms referring to the same node, Kirchoff's equation can then be rewritten as:

$$d\hat{V}_{outA} = d\hat{V}_{in} \frac{C_{MA}}{C_{tot}} + d\hat{V}_{outB} \frac{C_{MB}}{C_{tot}} + \frac{d\hat{I}}{C_{tot}} \quad (17)$$

where $C_{tot} = C_L + C_{MA} + C_{MB}$, whereas $d\hat{I} = qdX$.

This SDE can be used to model the state of the inverter in the presence of noise. Note that for circuits operated in sub-threshold with a total voltage swing of a few hundred mV, we can consider the transistor capacitances to be constant. The random electron counts, X_{ch} and X_{dis} , can be computed for arbitrarily complex CMOS gates. For the NAND shown

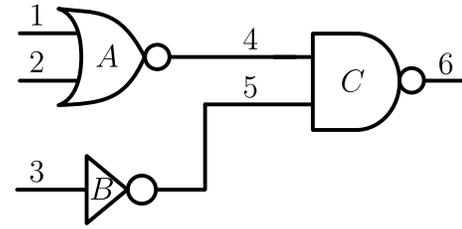


Fig. 5. Small example circuit used to describe the system of SDEs.

in Fig. 4, we can define $X_{ch} \sim \text{Pois}(\lambda_{p1} + \lambda_{p2} + \lambda_n)$ and $X_{dis} \sim \text{Pois}(\mu_{p1} + \mu_{p2} + \mu_n)$. Once again, the contribution of the bottom nMOS transistor to the pull-down transition rate comes from setting the voltage V_x , which determines the biasing point of the transistor connected to the output node. In our previous work [22], we built a discrete-time iterative function expressing the node voltage at each time step. This approach, however, has some limitations when the circuit under test has feedback paths, for example in ring oscillators and latches. In these cases, the iterative function has to be broken down, and the calculation is performed in two successive time steps. As we will see in Section V, this solution introduces an approximation that produces a drift between the predicted and the real transient response.

Instead, our new approach presented here uses a matrix-based description of the SDE system governing the circuit's transient behavior. We will use the small circuit in Fig. 5 to introduce the approach we use for building the system matrix. Following the same steps that led to (17), we can write an SDE for each node in the circuit, as follows:

$$\begin{aligned} d\hat{V}_4 C_{totA} - dV_1 C_{MA1} - dV_2 C_{MA2} - d\hat{V}_6 C_{MC1} &= d\hat{I}_A \\ d\hat{V}_5 C_{totB} - dV_3 C_{MB} - d\hat{V}_6 C_{MC2} &= d\hat{I}_B \\ d\hat{V}_6 C_{totC} - d\hat{V}_4 C_{MC1} - d\hat{V}_5 C_{MC2} &= d\hat{I}_C \\ C_{totA} &= C_{L4} + C_{MA1} + C_{MA2} + C_{MC1} \\ C_{totB} &= C_{L5} + C_{MB} + C_{MC2} \\ C_{totC} &= C_{L6} + C_{MC1} + C_{MC2}. \end{aligned}$$

Since V_1 , V_2 , and V_3 are external voltages, we can express the associated differential terms as known functions $f_1(t)$, $f_2(t)$, and $f_3(t)$, respectively, and move them to the right hand side of the equations. We can then write this system in matrix form as

$$\mathbf{C}d\hat{\mathbf{V}} = d\hat{\mathbf{I}}$$

where

$$\mathbf{d}\hat{\mathbf{V}} = \begin{bmatrix} d\hat{V}_4 \\ d\hat{V}_5 \\ d\hat{V}_6 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} C_{\text{tot}A} & 0 & -C_{MC_1} \\ 0 & C_{\text{tot}B} & -C_{MC_2} \\ -C_{MC_1} & -C_{MC_2} & C_{\text{tot}C} \end{bmatrix}$$

$$d\hat{\mathbf{I}} = \begin{bmatrix} d\hat{I}_A + f_1(t)C_{MA_1} + f_2(t)C_{MA_2} \\ d\hat{I}_B + f_3(t)C_{MB} \\ d\hat{I}_C \end{bmatrix}.$$

It is important to note that the system matrix is sparse for large logic circuits, which plays an important role in the performance scaling of our simulator, as will be discussed later. The elements in the vector $d\hat{\mathbf{I}}$ are the sums of the stochastic current components associated with the gate's transistors and, where applicable, the coupling current from the primary inputs.

Following the example we have shown in (7), the solution of this system of SDEs can be numerically computed. For instance, the solution at time t_n computed using the Euler method [45], which is the same method used in [22], can be written as

$$\mathbf{V}_n = \mathbf{V}_{n-1} + \mathbf{C}^{-1}d\hat{\mathbf{I}}_{n-1}. \quad (18)$$

Here, we have assumed that the matrix \mathbf{C} is invertible. This assumption can be justified considering that we model each node in the circuit with a lumped capacitance C_{tot} , which also makes all the diagonal entries nonzero values. While we have modified the mathematical framework used for describing the circuit, we have kept the noise-driven simulator approach originally presented in [22]. In contrast to other simulators that compute the stable solution and then add noise in the analysis, our formulation allows us to directly characterize the state of a CMOS circuit in the presence of thermal noise. As a result, we can greatly speed up the simulation time without sacrificing the accuracy of the results.

In order to add the effect of RTS noise to the formulation, we need to include the random current fluctuations produced by trapping and detrapping events in the definition of the thermal noise rates. Let us first extract the value of the equilibrium current from their stochastic representation. As mentioned in Section II, the equilibrium current values can be recovered from the expected values of the two Poisson distributions by using (8).

Without loss of generality, for the inverter in Fig. 1, we have

$$\begin{aligned} I_P - I_N &= q(\mathbf{E}[X_{\text{ch}}] - \mathbf{E}[X_{\text{dis}}]) \\ &= q(\lambda_{\text{inv}} - \mu_{\text{inv}}). \end{aligned} \quad (19)$$

For an nMOS transistor, the contribution of RTS noise to the total current can be written as

$$\begin{aligned} I + \Delta I &= I \left(1 + \frac{\Delta I}{I} \right) \\ &= I \left(1 + \sum_{i=1}^{N_{\text{tr}}} \frac{\Delta I}{I} \Big|_i \cdot \text{tr_state}_i \right) \\ &= q(\lambda - \mu) \left(1 + \sum_{i=1}^{N_{\text{tr}}} \frac{\Delta I}{I} \Big|_i \cdot \text{tr_state}_i \right) \end{aligned} \quad (20)$$

where tr_state is a variable equal to 1 if the trap is ‘‘filled’’ and 0 otherwise. By expanding (20), we can redefine the thermal and RTS noise rates as

$$\lambda_{\text{RTS}} = \lambda \left(1 + \sum_{i=1}^{N_{\text{tr}}} \frac{q}{m V_t W L C_{\text{ox}}} \left(1 - \frac{d_{\text{tri}}}{T_{\text{ox}}} \right) \cdot \text{tr_state}_i \right) \quad (21)$$

$$\mu_{\text{RTS}} = \mu \left(1 + \sum_{i=1}^{N_{\text{tr}}} \frac{q}{m V_t W L C_{\text{ox}}} \left(1 - \frac{d_{\text{tri}}}{T_{\text{ox}}} \right) \cdot \text{tr_state}_i \right). \quad (22)$$

This new analytical formulation allows us to include the effects of both RTS noise and thermal noise in the same set of Poisson rates for the charging and discharging processes.

IV. SIMULATOR IMPLEMENTATION

In the previous section, we have derived the SDE system used for computing the output voltage for a logic gate. Moreover, we have presented a new expression for the Poisson rates that incorporate the combined effects of RTS and thermal noise when computing the current fluctuation. In this section, we will first provide a description of the simulator implementation. This includes reviewing the approach presented in [22], highlighting the algorithmic modifications required to represent the system in matrix form, and describing the additional steps required to include RTS modeling in the framework. Next, we will describe our incremental threshold-crossing algorithm used to provide fast extraction of thermally induced SETs. In addition, we describe how the simulation time can be further improved through a multithreaded implementation.

In order to take into account the nonlinear response of the circuit to noise fluctuations, we need to update the rates of the Poisson processes at each time step, before sampling the random variables X_{ch} and X_{dis} . This updating is required because the rates depend exponentially on the biasing voltages V_{gs} and V_{ds} , as shown in (2) and (3). In addition, we need to know the values of the coupling capacitances and the total output node capacitance for each gate to build the system matrix \mathbf{C} .

A gate-level characterization of the Poisson rates and capacitances allows us to ensure accurate estimations at a minimum computational cost. For both pMOS and nMOS transistors, we build the rate look-up tables (LUTs), which can be indexed based on the V_{gs} and V_{ds} values. In particular, we precompute the rates from DC SPICE simulations by sweeping the values of both V_{gs} and V_{ds} with increments of 1 mV, which roughly corresponds to a single electron change in the

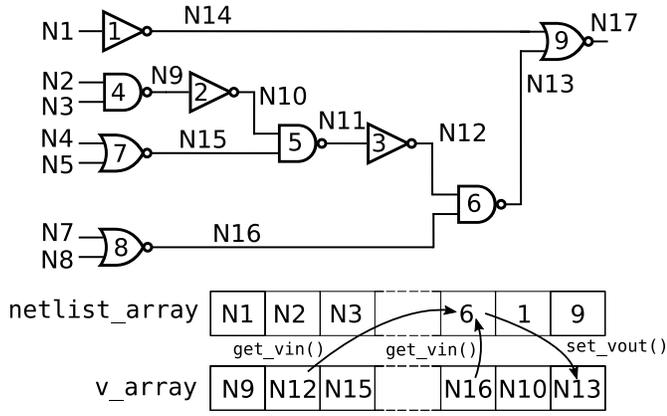


Fig. 6. Mapping the circuit netlist. `get_vin()` and `set_vout()` update the gate voltages at each iteration.

charge of a minimum-sized inverter in the considered technology, described by the 7 nm FinFET predictive technology model [37], [38].

Each gate in our library is defined as a separate class that contains the values of the internal capacitances and the methods for computing the total rate values from the rate LUTs. An abstract class representation allows us to conveniently rearrange the circuit netlist into an array of gate objects. The voltage value at each node is stored in a matrix of size $N \times \text{net_size}$, where N represents the depth of the simulation buffer and `net_size` is the number of nodes in the circuit. A virtual connection between the gates is achieved by storing the appropriate indexes to the voltage matrix as input and output indexes. Fig. 6 shows a simple example netlist mapped into a `netlist_array` which contains the gate instances, and a `v_array` that stores the indexed voltage values for each node. The functions `get_vin()` and `set_vout()` are used to set the biasing point for each gate.

During the netlist initialization, we associate a total output capacitance of each node as the sum of its drain capacitance and the gate capacitance of every input connected to it. Similarly, we build the system matrix \mathbf{C} by assigning the ratio between the fanout capacitance from node j and the total capacitance for node i to c_{ij} . Here, we have assumed that the gates in the netlist are ordered following the circuit causality, which we ensure when parsing the Verilog input file. For the matrix representation, we used the Eigen library [46], which provides classes and methods for linear sparse problem solving. Finally, we perform the RTS trap profiling following the methodology from [32], and the parameters summarized in Table I. We initialize the traps for each gate with a random state (either filled or “empty”) and a next event time. The list of traps and the correspondent times are then sorted in increasing event time order. The amplitude of the RTS noise is computed by adding together the contribution of each trap in the filled state. This initialization procedure is described in the `Init` block of Algorithm 1.

The simulation core is shown in the `Sim` block of Algorithm 1. Given a simulation time of length T_{sim} , we divide our iterations in time windows of length $T_N = T_{\text{sim}}/N$. This allows us to optimize the simulation runtime by sampling the

Algorithm 1: Simulator Algorithm Description. `Init` Parses the Input Verilog to Create a Netlist, and Initializes the Gates’ Parameters. At This Stage, the System Matrix Is Populated As Well. `Sim` Executes the Actual Noise Transient Simulations. The Algorithm Solves the SDE System and Computes the Derivatives Using Euler’s Numerical Method

Data: Verilog structural netlist, transistor rates LUTs
Result: Voltage time-series

```

Init Parse Verilog file
Create netlist_array of gate objects
Initialize output matrix v_array
foreach gate_i in netlist_array do
    gate_i.Cout ← gate_i.Cdd
    foreach gate_j in netlist_array \gate_i do
        if gate_i.O_idx is equal to gate_j.I_idx then
            gate_i.Cout ← gate_i.Cout + gate_j(Cgg)
            add gate_j to gate_i.Fan_out
    Initialize gate_i.rts_traps
    foreach trap in gate_i.rts_traps do
        add get_next_time(trap.rates) to gate_i.rts_times
    Sort gate_i.rts_traps, gate_i.rts_times
Populate System Matrix C
Sim n = 0
N = T_sim/T_N
while n < N do
    foreach gate_i in netlist_array do
        if t > gate_i.rts_times[0] then
            rts_thinning(gate_i.rts_traps[0])
            add get_next_time(gate_i.rts_traps[0].rates) to
            gate_i.rts_times
            Sort gate_i.rts_traps, gate_i.rts_times
    for t ∈ {0, ..., T_N} do
        foreach gate_i in netlist_array do
            gate_i.get_vin()
            gate_i.set_vout()
            Update parameters λ_t, μ_t
            X_t ← gate_i.gen_noise(λ_t, μ_t)
            dÎ[i] = X_t × q
            if gate_i.V_in in PI then
                dÎ[i] = dÎ[i] +
                    (gate_i.V_in(t) - gate_i.V_in(t - 1)) × gate_i.C_M
            V_t = V_t - 1 + C.solve(dÎ)
        n ++

```

RTS and thermal noise fluctuations on different time scales. In fact, while thermal noise is evaluated at each time step, which is in the order of 10 ps, the RTS noise process does not need the same time resolution, as the characteristic time constant of the traps is in the ms time scale (see Fig. 2). As a result, using (21) and (22), we can include the effect of RTS current shifts in the rates responsible for thermal noise instead of treating the two sources independently.

In order to run the RTS model in the time domain, we use an approach similar to Aadithya *et al.* [33], [34]. At the beginning of each iteration window, for each gate, we select the trap

Algorithm 2: Algorithm Describing the Generation of RTS Trapping and Detrapping Events. The Algorithm Is Called at the Beginning of Each Simulation Window, T_N , and it Uses a Thinning Algorithm for Generating Nonhomogeneous Poisson Samples

Data: A RTS trap

Result: The updated RTS state sampled from the NHPP

$t_{next} \leftarrow 0$

init trap_state

$u \leftarrow \mathcal{U}(0, 1)$

$t_{next} \leftarrow t_{next} - \frac{\ln(u)}{\lambda^*}$

At beginning of simulation window

if $t_{next} \geq t$ **then**

if trap_state is “filled” **then**

$\lambda_t \leftarrow \lambda_e$

else

$\lambda_t \leftarrow \lambda_c$

$v \leftarrow \mathcal{U}(0, 1)$

if $v < \frac{\lambda_t}{\lambda^*}$ **then**

 trap_state \sim trap_state

$u \leftarrow \mathcal{U}(0, 1)$

$t_{next} \leftarrow t_{next} - \frac{\ln(u)}{\lambda^*}$

with the earliest event time and we update the trap status due to RTS noise according to a nonhomogeneous Poisson process (NHPP). The pseudocode for computing samples from an NHPP is summarized in Algorithm 2. The function generates interarrival times from an exponential distribution with rate defined by the maximizing rate λ^* . Given the current rate λ_t , each of these events are then captured with probability (λ_t/λ^*) . In the case of our implementation, the candidates for the next event time are computed from an exponential distribution with rate $\lambda^* > \max((1/\tau_e), (1/\tau_c))$. Note that this parameter depends only on the maximum rate and not on the current state of the traps. At the beginning of each time window of length T_N , we check if we have reached the next event candidate for each trap and we change the state with probability $(\lambda_{\{e,c\}}/\lambda^*)$. Whenever we reach a transition time, regardless of the sampling outcome, we also compute a new candidate for the next event time. Fig. 7 shows an example of the output generated by the RTS noise algorithm.

Once we have completed the RTS procedure, we proceed with simulating the circuit transient evolution for a fixed time step Δt , which we choose based on the fastest time constant for a minimum-sized inverter. For $V_{DD} = 180$ mV, we fixed the time step at 50 ps. In these conditions, we can assume that the rates do not change markedly within a single time step. Therefore, we compute the noise fluctuations at each time step without resorting to thinning. The resulting random fluctuation X_t is used to update the values in the $\hat{\mathbf{I}}$ vector. The algorithm for computing thermal noise fluctuations at each time step, is shown in Algorithm 3.

The derivatives are computed using Euler’s method as shown in (18). Note that we use the inverse matrix just for sake of notation. Instead, the value $\mathbf{C}^{-1}\hat{\mathbf{I}}$ is computed using

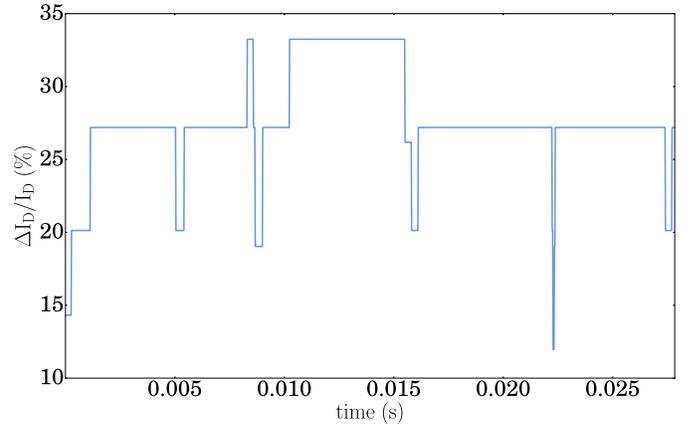


Fig. 7. RTS noise extracted from our simulator. The figure shows the relative current noise $(\Delta I/I)$ for a device with several traps. For the 7 nm FinFET model considered in this paper, the maximum noise for a single trap at the Si–HfO₂ interface, is 9%.

Algorithm 3: Algorithm for Generating Samples From the Charging and Discharging Poisson Processes

Data: The current rates λ_t and μ_t

Result: A sample X_t from a Poisson process in a time interval Δt

$t_0 \leftarrow 0$

event_fw $\leftarrow 0$

event_rev $\leftarrow 0$

$u \leftarrow \mathcal{U}(0, 1)$

$t_0 \leftarrow t_0 - \frac{\ln(u)}{\lambda_t}$

while $t_0 < \Delta t$ **do**

 event_fw++

$u \leftarrow \mathcal{U}(0, 1)$

$t_0 \leftarrow t_0 - \frac{\ln(u)}{\lambda_t}$

$t_0 \leftarrow 0$

$u \leftarrow \mathcal{U}(0, 1)$

$t_0 \leftarrow t_0 - \frac{\ln(u)}{\mu_t}$

while $t_0 < \Delta t$ **do**

 event_rev++

$u \leftarrow \mathcal{U}(0, 1)$

$t_0 \leftarrow t_0 - \frac{\ln(u)}{\mu_t}$

$X_t \leftarrow \text{event_fw} - \text{event_rev}$

one of the sparse solvers available in the Eigen library, which allows us to take advantage of the sparse nature of the system matrix for speeding up the simulation.

A. Fast Extraction of Thermally Induced SET

As will be explained in Section V, the proposed implementation allows us to speed up noise transient simulations by at least three orders of magnitude when compared to SPICE-based simulations. This translates to the ability of detecting up to 7σ events in hours rather than days, where $\sigma = \sqrt{kT/C}$. However, we are interested in capturing rare fault-inducing transients, whose deviation from the mean is greater than 7σ . Therefore, we propose a methodology to

reduce the simulation time of thermal noise transients even further, based on the use of an incremental threshold-crossing algorithm. The main principle behind this algorithm follows from the Markov property associated with our simulation framework. In fact, the net electron count at the output node of each CMOS gate can be modeled as a Markov chain with transition probabilities determined by the two Poisson distributions governing the charging and discharging processes. As such, the next state of the system, i.e., the charge on the output capacitance at the next time step, depends only on the present state and not on the sequence of voltage fluctuations that brought the node to its current state.

The Markov property can be used to build a set of Monte-Carlo simulations in which the voltage at a given node will progressively approach a large deviation event, without incurring the computational cost of a very large number of iterations. As an example, let us consider the initial condition in which all the nodes in the circuit are at equilibrium, i.e., depending on the logic gate that drives them, their voltage will be set at either V_{DD} or 0. We shall label this initial state as S_0 . At this point, we start running a transient simulation looking for a deviation from the equilibrium condition that can be reached using the approach shown in the previous section. We label the voltage value associated with this event as $V_\sigma[0]$. At each time step, we check whether any of the circuit node voltages are within the region defined by the inequality $V_\sigma[0] < V_{out_i} < (V_{DD} - V_\sigma[0])$, indicating a departure from the equilibrium logic value of a node. Once we have reached this event for any of the nodes in the circuit, we save the event time and the voltages for all nodes as a new initial state, S_1 , which we set as the initial condition for a new simulation run. Let us define the event time as $\text{period_count} \times T_N + t_{\text{event}}$, where period_count is the number of times we loop through the simulation window, T_N is the size of the simulation window, and $0 \leq t_{\text{event}} < T_N$ is the time step within the simulation window at which we reached the threshold-crossing condition. Starting from the initial state S_1 , we seek an event deviating from the mean by $V_\sigma[1] > V_\sigma[0]$. As we move further away from equilibrium, we can expect that the vast majority of the simulation outcomes will return back to the equilibrium point rather than push the voltage toward larger deviations. Our algorithm detects these outcomes within a few time steps, at which point, the simulation is interrupted and reverted to the latest valid initial state.

As an example, let us assume that we reached the event $V_\sigma[n]$ and the simulation has been reset to the state S_{n+1} associated with the n th threshold crossing. As the transient simulation progresses, we have three possible outcomes.

- 1) We fall back below the threshold at $V_\sigma[n-1]$. In this condition we can assume that the noise-driven excursion is likely to go back to the equilibrium conditions and therefore we exit the current run, increment an exit counter and restart the simulation at state S_{n+1} .
- 2) We reach the next threshold $V_\sigma[n+1]$. As for the initial case, we save the current time t_{event} , and the current state S_{n+2} as next reset state.
- 3) We reach the end of the simulation buffer without leaving the boundaries defined by $V_\sigma[n-1]$ and $V_\sigma[n+1]$ in

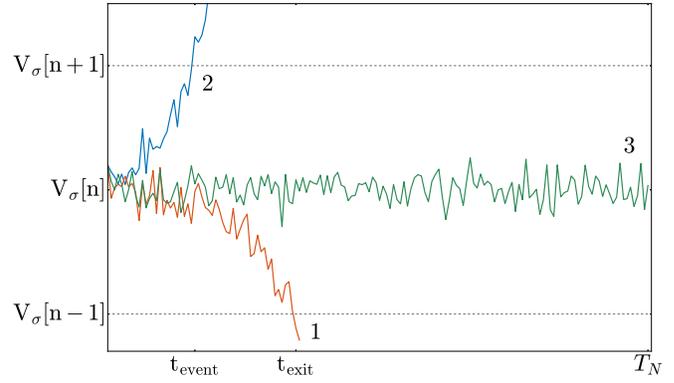


Fig. 8. Graphical representation of the three possible scenarios between iterations of the iterative threshold-crossing algorithm. Note that waveform 1 represents the dominant outcome.

Algorithm 4: Pseudo-Code for the Iterative Threshold Crossing Algorithm. The Algorithm Performs the Noise Transient Simulation Until the Output Voltage of One of the Gates in the Circuit Exceeds the Final Desired Value, $V_\sigma[K-1]$

Data: An array of K incremental voltage thresholds
 $V_\sigma[0] \dots V_\sigma[K-1]$

Result: The transient voltage matrix V , containing a SET
 $n = 0$

while $gate_i.V_{out} < V_\sigma[K-1]$ **for all** $gate_i$ **in**
 $netlist_array$ **do**

for t **in** $(0, \dots, T_N)$ **do**

foreach $gate_i$ **in** $netlist_array$ **do**

if $gate_i.V_{out}(t) > V_\sigma[n+1]$ **then**

$S_{n+2} \leftarrow V_t$

$n++$

continue

else if $gate_i.V_{out}(t) < V_\sigma[n-1]$ **then**

$t = 0$

$V_t \leftarrow S_{n+1}$

$exit_count++$

continue

$period_count++$

which case we increment a period counter and continue the simulation using the last available state.

The three possible outcomes are illustrated in Fig. 8, and the algorithm's pseudocode is shown in Algorithm 4. Notice that, while we check for threshold-crossing events for all the gates in the circuit, this search is interrupted as soon as the next threshold is reached by any of the gates, and the algorithm keeps running until the output of at least one gate reaches the final desired value $V_\sigma[K-1]$.

Using this algorithm, we lose the sequential structure of the stochastic time-domain simulation. However, given the recursive approach for building the noise traces between thresholds, we can retrieve the time to reach each of the crossing events using the following equation:

$$T[n+1] = T[n] + \text{exit_count} \times \bar{T}[n-1] + \text{period_count} \times T_N + t_{\text{event}}. \quad (23)$$

The term $\text{exit_count} \times \bar{T}[n-1]$ takes into account that every time we exit the current run we assume that the process will return to the equilibrium condition and it will take on average a time $\bar{T}[n-1]$ to return to the same state. Since this event is usually within a few samples from the beginning of the simulation, we can immediately discard these runs and hence achieve a considerable gain in computation time.

We conclude this discussion with two considerations. First, as our approach requires dividing the voltage amplitude into incremental steps $V_\sigma[0] \cdots V_\sigma[N]$, it is important to identify a criterion for selecting these thresholds. A sensible way to divide up this voltage range is to use the noise standard deviation at equilibrium, $\sqrt{kT/C}$. However, in order to reduce the overhead associated with checking the threshold-crossing condition for small deviations, we set the initial threshold to a value between $3 \times \sqrt{kT/C}$ and $4 \times \sqrt{kT/C}$, while spacing the following thresholds by $\sim \sqrt{kT/C}$. Second, it is important to clarify that all these simulations are run for steady-state input voltages, which allows us to isolate the effect of noise from transients caused by changes in the primary input signals that propagate through the circuit.

B. Multithreaded Implementation

The algorithm we just outlined can be further optimized through parallelization. In general, circuit simulators take advantage of multithreading by parallelizing the population and inversion of the nodal matrix [47]. In our case, the most time consuming operation consist of sampling over the possible circuit transient simulations in order to find the next threshold-crossing event. Therefore, a straightforward way to improve the simulation performance is to run the event search in parallel. In order to accomplish this, we keep a voltage matrix for each thread. Moreover, since each thread will have to set different input and output voltages for the gates in the circuit, we assign a copy of the netlist to each thread. The netlist copy is performed by adding a clone() function to the gate classes. In order to keep consistency between all threads, each thread should run the simulation starting from the same state. This requires initializing the voltage matrix for each thread with the value of the last detected event.

A description of the whole simulation flow is depicted in Fig. 9. During the initialization process, we extract the Poisson rates from the transistor I-V curves at a given temperature and operating voltage, and initialize the circuit netlist and voltage matrix. After the simulation setup is completed, we start the simulation. Each thread independently generates the noise samples based on Algorithm 1 and checks if any of the voltage nodes have reached the first threshold $V_\sigma[0]$. Once an event is detected by one of the threads, the remaining threads are locked. The thread that reached the event writes back the data, reinitializes the voltage matrix for all threads to its current state, and sets an exit flag that forces all threads to resume the simulation from the reset state. This operation continues until the final threshold is reached.

V. RESULTS

As a first set of experiments, we want to evaluate how well our simulator can accurately model transient effects. As an

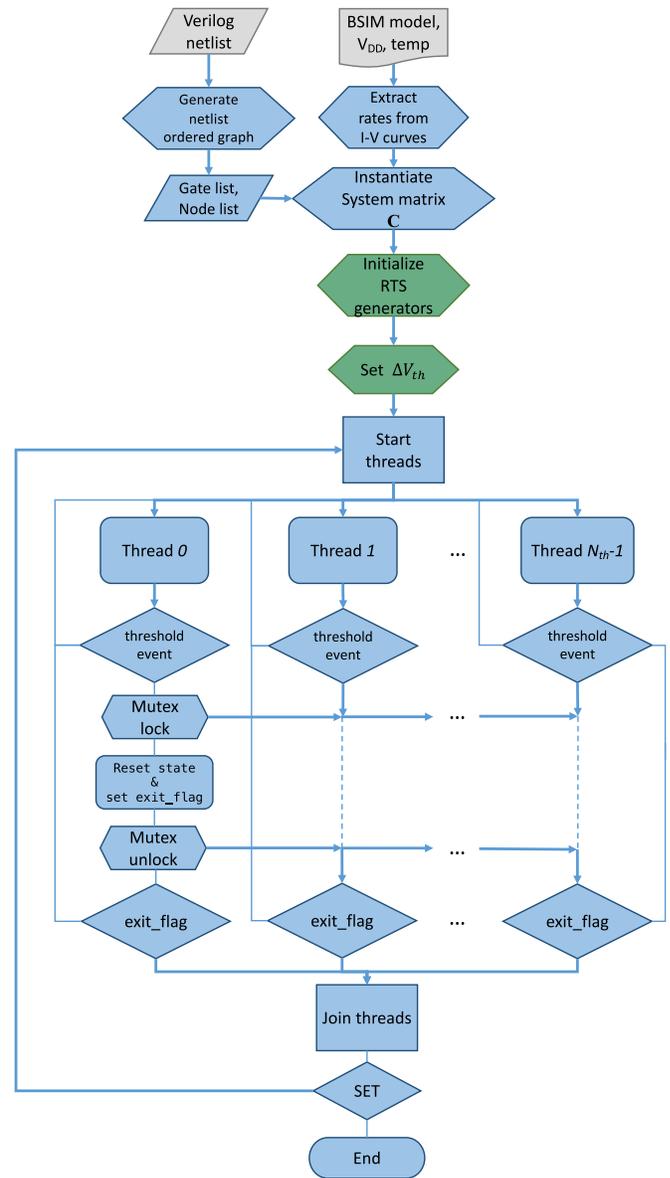


Fig. 9. Multithreaded simulation flow: the setup first requires extracting the Poisson rates from SPICE I-V simulations for given temperature and operating voltages. Additionally, the circuit topology is derived from verilog structural netlists. The noise simulation is performed by each thread independently. In the provided example, a crossing event is detected by thread 0.

example, we show in Fig. 10 the output of a NAND gate driving an inverter from our simulator as compared to SPICE. The two traces are accurately matched for both under/overshoot and tail effects.

In order to ascertain the runtime improvement of our approach, we compared the runtime of our simulator for a full $100 \mu\text{s}$ simulation against a SPICE transient noise analysis. For SPICE, we ran the simulation for shorter times and extrapolated a linear trend, which led to the estimated runtime for a $100 \mu\text{s}$ simulation. The simulations were run over a set of circuits from the MCNC benchmark suite and results are summarized in Table II. All circuits were simulated at $T = 100 \text{ }^\circ\text{C}$ and $V_{DD} = 180 \text{ mV}$. Note that this chosen value for the supply voltage is in line with other works that target noise-immune design for sub-threshold circuits [4], [23], [26].

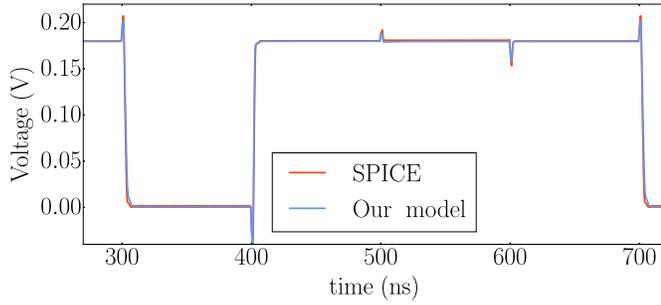


Fig. 10. Output of a NAND gate as generated from our simulator and SPICE.

TABLE II
 RUNTIME OF OUR SIMULATOR FOR MCNC BENCHMARK CIRCUITS FOR A NOISE TRANSIENT SIMULATION OF $100\ \mu\text{s}$. THE SPEEDUP IS MEASURED AGAINST THE ESTIMATED RUNTIME OF SPICE. FOR LARGER CIRCUITS, THIS PAPER HAS BETTER PERFORMANCE COMPARED TO [22], THANKS TO THE ADOPTION OF A SPARSE SOLVER

	# gates	# PI	# PO	[22] [minutes]	This work [minutes]	SPICE [days]	speedup
rd53	63	5	3	6.1	6.48	5.6	1245
b9	125	41	21	11.9	12.01	14.6	1750
9sym	254	9	1	28.2	26.96	29	1549
rd84	263	8	4	27.4	26.74	25.8	1389
apex2	452	39	3	49.2	47.81	45	1355
amd	629	14	24	70	69.11	68	1417
ex5	907	8	63	101.7	98.88	92	1340
vda	1021	17	39	162.5	111.92	126	1621
t481	1467	16	1	291.6	162.28	133	1180
seq	2258	41	35	346	254.54	368.8	2086

We tested circuits with up to 2258 gates and achieved a speedup of three orders of magnitude. Using a sparse solver for the solution of the SDE system increases the performance of the simulator for larger circuits compared to [22].

While our simulator can greatly improve the simulation runtime just by running a standard transient simulation, we extended its functionality using the methodology described at the end of Section IV to improve the ability of capturing rare transients. We tested this algorithm on a chain of 16 inverters and measured the time to event for thresholds up to 40 mV, which is the maximum amplitude for which we could gather samples for the standard SPICE transient simulation in a reasonable time.

In order to properly extrapolate the statistical event time distribution, we completed a set of 1000 runs for each event threshold. The time to reach the event follows an exponential distribution, as shown in Fig. 11. For the iterative approximation, we used the same interval $V_\sigma = 20 - 40\ \text{mV}$ with steps $\Delta V_\sigma = 5\ \text{mV}$. When we apply our iterative threshold-crossing algorithm for the extraction of SETs, the distribution is skewed and can be fitted using a log-normal distribution. This behavior follows a particular case of the central limit theorem (CLT). Specifically, for higher thresholds, the predominant factor in (23) is given by $\text{exit_count} \times \bar{T}[n-1]$. Therefore, successive event times will be generated by multiplying together several independent random variables. Given the natural logarithm of these values, the CLT states that the sum of the

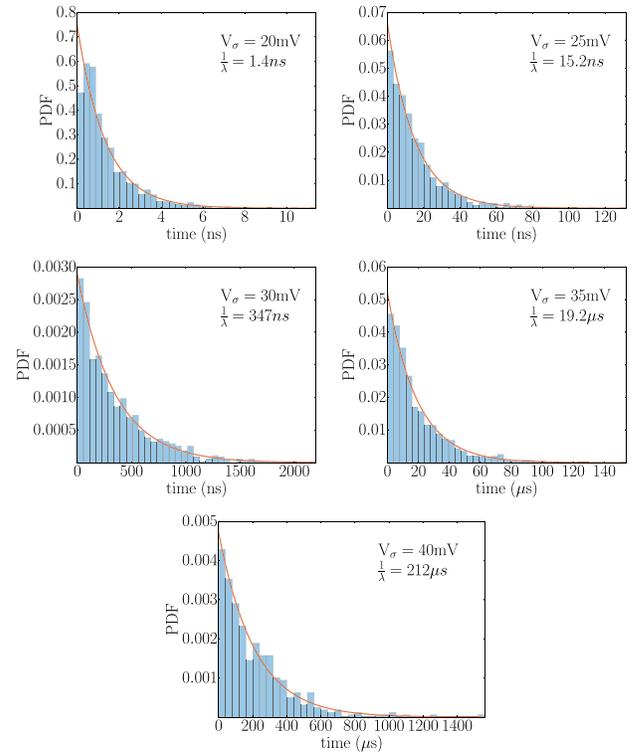


Fig. 11. Statistical distribution for five time-to-threshold transient simulations using values from 20 to 40 mV. The data was fitted using an exponential probability density function.

logarithms, will follow a Gaussian distribution, and therefore, the original samples' distribution will be log-normal. The time-to-event sample distributions are plotted in Fig. 12. It is worth pointing out that the distribution for the first two thresholds, 20 and 25 mV still follows an exponential distribution. This is not surprising as for the first two iterations of the iterative threshold-crossing algorithm, the exit condition is never met. The comparison between the mean time for a simple transient run, following an exponential distribution, and the higher moments associated with the log-normally distributed samples is shown in Fig. 13. This technique could generate SET reaching amplitudes over 12σ with an average runtime of 56.2 s. Fig. 14 shows a sample SET from our simulator with a peak amplitude of 71 mV. It is worth mentioning that we noticed that the overhead for creating the threads would be too costly for those thresholds that could be reached within a few seconds. Therefore, we enable the multithreaded simulation only when the signal amplitude is greater than 30 mV, when we start observing exit events. The extrapolated time to event was in the order of 10^{16} s which would be clearly infeasible to simulate with traditional transient simulation approaches. If we wanted to look for a yardstick for this measure, it is useful to keep in mind that the age of the universe is estimated at 1.41×10^{17} s. The large estimate for the time-to-failure should be put into perspective considering that large modern circuits have a much higher number of transistors compared to the simulated circuit. Therefore, we can expect the error rate to increase with the number of gates.

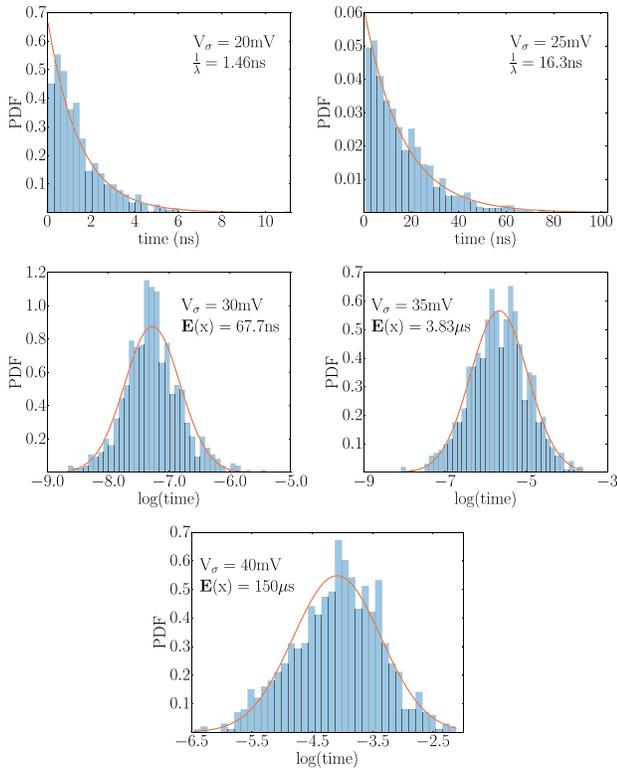


Fig. 12. Statistical distribution for the same time-to-threshold transients as Fig. 11 computed using the threshold-crossing approach.

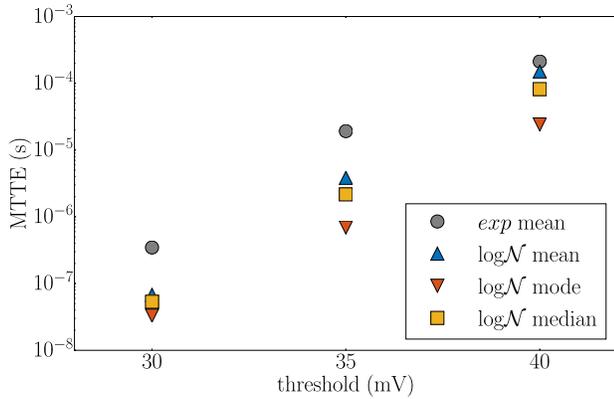


Fig. 13. Comparison between the mean-time-to-event from a single run (exponential distribution), and the higher moments for samples computed using the iterative threshold-crossing method.

In addition to the analysis of thermally induced SETs, we performed a study of the impact of RTS noise on logic circuits. For this purpose, we evaluated the impact of both RTS and thermal noise on the clock jitter in a ring oscillator [48]. As we mentioned in Section III, one of the reasons for switching to a matrix-based solution for the SDE system is that the two-step method proposed in [22] requires to push the computation of feedback components to the next iteration. In the previous sections of this paper, we have considered only combinational logic, for which the only feedback components that appear in the SDEs are the coupling currents from the fanout stages. For those cases, the implementation

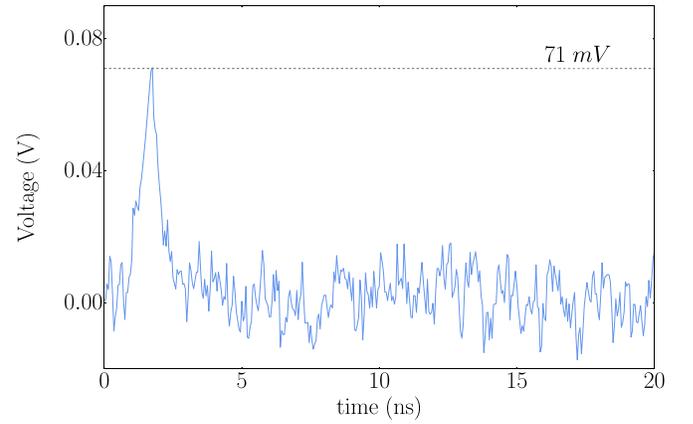


Fig. 14. Example of SET generated by our simulator.

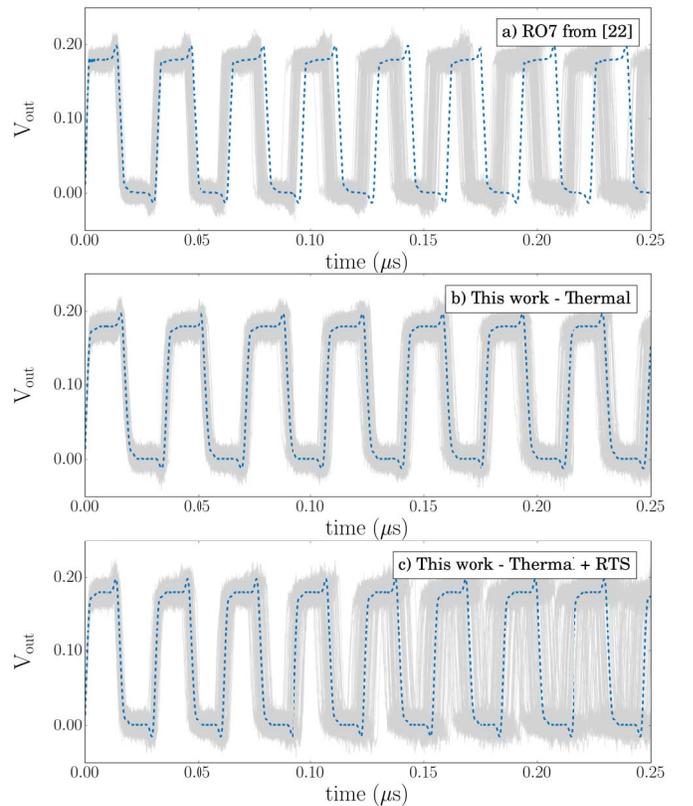


Fig. 15. Simulation of a 7-stage ring oscillator using [22] (a), and this paper, (b) and (c). Computing the solution of the SDE in two different iterations produces a drift in the noisy simulation compared to the noiseless one.

proposed in [22] offers good accuracy in evaluating the transient response. However, the limitations of that solution are particularly marked when simulating ring oscillators. Fig. 15 shows the result of simulating a 7-stage ring oscillator using the two different approaches. In both cases, the injection of noise causes variations in the oscillator transition edges in the output voltage. Notice that, although most traps are unlikely to change their state in the small simulation time of $0.25 \mu\text{s}$, we can still evaluate the effect of the initial random trap characterization at the beginning of the simulation. Fig. 15(a) shows the output of the ring oscillator computed using a two-step

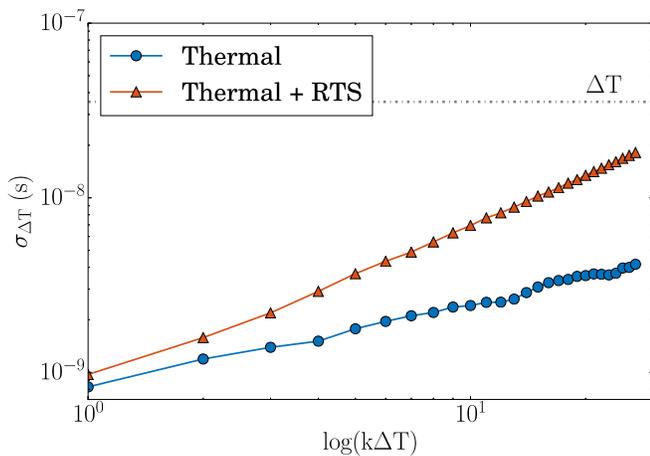


Fig. 16. Standard deviation of the clock jitter computed at each integer multiple of the clock period, $T \sim 35$ ns, for thermal, and thermal and RTS noise.

solution for the SDE. This solution induces a drift that is amplified when the noise is injected in the circuit. On the other hand, the matrix-based implementation produces oscillations that are centered around the mean value, represented by the dashed waveform. This result is shown in Fig. 15(b) and (c), which illustrate the effect of thermal noise only and thermal and RTS noise combined. In the latter case, the clock jitter causes the overlap of edges from adjacent periods after a few cycles. A quantitative measure of the clock jitter is given by the standard deviation of the distribution of zero-crossing times for the noisy circuit, as shown in Fig. 16. In both cases, the value of the clock jitter rapidly approaches the clock period ΔT .

VI. CONCLUSION

In this paper, we have described the approach for designing our fast circuit simulator for the analysis of thermal and RTS noise transients in sub-threshold circuits. The SDE-based implementation allows us to efficiently model transient effects and run simulations independently from SPICE. The results presented in this paper show that our simulator performs three orders of magnitude faster than SPICE-based simulations. Moreover, the matrix-based implementation provides increased accuracy and better scalability with the circuit size compared to our previous work. In addition to these improvements, we have introduced a Monte-Carlo simulation approach that can accurately extract high-amplitude SETs. We have demonstrated that our iterative multithreshold algorithm can be used to capture extremely rare noise-induced transient events in times that would be impossible to reach with standard transient simulators. At the same time, the analysis of the mean-time-to-event has shown that intrinsic noise sources alone are unlikely to endanger the correct functionality of relatively small combinational logic circuits considered to date. This may change in the future, as logic circuits based on minimum-sized transistors come to exceed billion-gate counts.

ACKNOWLEDGMENT

The authors would like to thank Prof. C.-W. Shu for his insightful suggestions in developing the matrix-based

description of the SDE system and the anonymous referees for their very thorough and helpful comments. Part of this paper was conducted using computational resources and services at the Center for Computation and Visualization, Brown University.

REFERENCES

- [1] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 6, pp. 2874–2878, Dec. 1996.
- [2] D. R. Ball *et al.*, "Simulating nuclear events in a TCAD model of a high-density SEU hardened SRAM technology," in *Proc. RADECS*, 2005, pp. C12-1–C12-5.
- [3] M. Zhang *et al.*, "Sequential element design with built-in soft error resilience," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 12, pp. 1368–1378, Dec. 2006.
- [4] J. P. Kulkarni, K. Kim, and K. Roy, "A 160 mV robust Schmitt trigger based subthreshold SRAM," *IEEE J. Solid-State Circuits*, vol. 42, no. 10, pp. 2303–2313, Oct. 2007.
- [5] S. Almukhaizim and Y. Makris, "Soft error mitigation through selective addition of functionally redundant wires," *IEEE Trans. Rel.*, vol. 57, no. 1, pp. 23–31, Mar. 2008.
- [6] D. B. Limbrick, N. N. Mahatme, W. H. Robinson, and B. L. Bhuva, "Reliability-aware synthesis of combinational logic with minimal performance penalty," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 4, pp. 2776–2781, Aug. 2013.
- [7] N. Miskov-Zivanov and D. Marculescu, "MARS-S: Modeling and reduction of soft errors in sequential circuits," in *Proc. ISQED*, 2007, pp. 893–898.
- [8] S. Mitra *et al.*, "Combinational logic soft error correction," in *Proc. ITC*, Santa Clara, CA, USA, 2006, pp. 1–9.
- [9] R. R. Rao, D. Blaauw, and D. Sylvester, "Soft error reduction in combinational logic using gate resizing and flipflop selection," in *Proc. ICCAD*, 2006, pp. 502–509.
- [10] B. D. Sierawski, B. L. Bhuva, and L. W. Massengill, "Reducing soft error rate in logic circuits through approximate logic functions," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 6, pp. 3417–3421, Dec. 2006.
- [11] K.-C. Wu and D. Marculescu, "A low-cost, systematic methodology for soft error robustness of logic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 2, pp. 367–379, Feb. 2013.
- [12] *ITRS Executive Summary*, ITRS, 2013. [Online]. Available: <http://www.itrs2.net/itrs-reports.html>
- [13] B. H. Calhoun and A. Chandrakasan, "Characterizing and modeling minimum energy operation for subthreshold circuits," in *Proc. ISLPED*, 2004, pp. 90–95.
- [14] B. H. Calhoun, S. Khanna, R. Mann, and J. Wang, "Sub-threshold circuit design with shrinking CMOS devices," in *Proc. ISCAS*, 2009, pp. 2541–2544.
- [15] H. Kaul *et al.*, "Near-threshold voltage (NTV) design: Opportunities and challenges," in *Proc. DAC*, 2012, pp. 1149–1154.
- [16] L. B. Kish, "End of Moore's law: Thermal (noise) death of integration in micro and nano electronics," *Phys. Lett. A Gen. Atomic Solid State Phys.*, vol. 305, nos. 3–4, pp. 144–149, 2002.
- [17] J. P. Campbell *et al.*, "Large random telegraph noise in sub-threshold operation of nano-scale nMOSFETs," in *Proc. ICICDT*, Austin, TX, USA, 2009, pp. 17–20.
- [18] A. G. Mahmutoglu and A. Demir, "Modeling and simulation of low-frequency noise in nano devices: Stochastically correct and carefully crafted numerical techniques," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 5, pp. 794–807, May 2015.
- [19] G. Wirth *et al.*, "Compact modeling and simulation of random telegraph noise under non-stationary conditions in the presence of random dopants," *Microelectron. Rel.*, vol. 52, no. 12, pp. 2955–2961, 2012.
- [20] Y. Ye, C.-C. Wang, and Y. Cao, "Simulation of random telegraph noise with 2-stage equivalent circuit," in *Proc. ICCAD*, San Jose, CA, USA, 2010, pp. 709–713.
- [21] R. Sarpeshkar, T. Delbruck, and C. A. Mead, "White noise in MOS transistors and resistors," *IEEE Circuits Devices Mag.*, vol. 9, no. 6, pp. 23–29, Nov. 1993.
- [22] M. Donato, R. I. Bahar, W. Patterson, and A. Zaslavsky, "A fast simulator for the analysis of sub-threshold thermal noise transients," in *Proc. DAC*, Austin, TX, USA, 2016, pp. 1–6.

- [23] M. Donato *et al.*, "A noise-immune sub-threshold circuit design based on selective use of Schmitt-trigger logic," in *Proc. GLSVLSI*, Salt Lake City, UT, USA, 2012, pp. 39–44.
- [24] L. García-Leyva, A. Calomarde, F. Moll, and A. Rubio, "Novel redundant logic design for noisy low voltage scenarios," in *Proc. LASCAS*, Cusco, Peru, 2013, pp. 1–4.
- [25] B. Marr, J. George, B. Degnan, D. V. Anderson, and P. Hasler, "Error immune logic for low-power probabilistic computing," *VLSI Des.*, vol. 2010, pp. 1–10, Jan. 2010.
- [26] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky, "Designing logic circuits for probabilistic computation in the presence of noise," in *Proc. DAC*, Anaheim, CA, USA, 2005, pp. 485–490.
- [27] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- [28] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Berlin, Germany: Springer-Verlag, 1992.
- [29] H. Li *et al.*, "A model for soft errors in the subthreshold CMOS inverter," in *Proc. SELSE*, 2006, pp. 1–4.
- [30] P. Jannaty *et al.*, "Shot-noise-induced failure in nanoscale flip-flops part II: Failure rates in 10-nm ultimate CMOS," *IEEE Trans. Electron Devices*, vol. 59, no. 3, pp. 807–812, Mar. 2012.
- [31] M. J. Uren, D. J. Day, and M. J. Kirton, "1/f and random telegraph noise in silicon metal-oxide-semiconductor field-effect transistors," *Appl. Phys. Lett.*, vol. 47, no. 11, p. 1195, 1985.
- [32] M. V. Dunga, "Nanoscale CMOS modeling," Ph.D. dissertation, EECS Dept, Univ. California at Berkeley, Berkeley, CA, USA, 2008.
- [33] K. V. Aadithya, A. Demir, S. Venugopalan, and J. Roychowdhury, "SAMURAI: An accurate method for modelling and simulating non-stationary random telegraph noise in SRAMs," in *Proc. DATE*, 2011, pp. 1–6.
- [34] K. V. Aadithya, S. Venugopalan, A. Demir, and J. Roychowdhury, "MUSTARD: A coupled, stochastic/deterministic, discrete/continuous technique for predicting the impact of random telegraph noise on SRAMs and DRAMs," in *Proc. DAC*, 2011, pp. 292–297.
- [35] A. R. Trivedi *et al.*, "A simulation study of oxygen vacancy-induced variability in HfO₂/metal gated SOI FinFET," *IEEE Trans. Electron Devices*, vol. 61, no. 5, pp. 1262–1269, May 2014.
- [36] X. Sun *et al.*, "Study of gate oxide traps in HfO₂/AlGaIn/GaN metal-oxide-semiconductor high-electron-mobility transistors by use of ac transconductance method," *Appl. Phys. Lett.*, vol. 102, no. 10, 2013, Art. no. 103504.
- [37] N. Paydavosi *et al.*, "BSIM—SPICE models enable FinFET and UTB IC designs," *IEEE Access*, vol. 1, pp. 201–215, 2013.
- [38] S. Sinha, G. Yeric, V. Chandra, B. Cline, and Y. Cao, "Exploring sub-20nm FinFET design with predictive technology models," in *Proc. DAC*, San Francisco, CA, USA, 2012, pp. 283–288.
- [39] G. Kapila, B. Kaczer, A. Nackaerts, N. Collaert, and G. V. Groeseneken, "Direct measurement of top and sidewall interface trap density in SOI FinFETs," *IEEE Electron Device Lett.*, vol. 28, no. 3, pp. 232–234, Mar. 2007.
- [40] J. W. Lee *et al.*, "Low frequency noise analysis for post-treatment of replacement metal gate," *IEEE Trans. Electron Devices*, vol. 60, no. 9, pp. 2960–2962, Sep. 2013.
- [41] S. Monaghan, P. K. Hurley, K. Cherkaoui, M. A. Negara, and A. Schenk, "Determination of electron effective mass and electron affinity in HfO₂ using MOS and MOSFET structures," *Solid State Electron.*, vol. 53, no. 4, pp. 438–444, 2009.
- [42] R. Talmat *et al.*, "Low frequency noise characterization in n-channel FinFETs," *Solid State Electron.*, vol. 70, pp. 20–26, Apr. 2012.
- [43] Y.-C. Yeo, T.-J. King, and C. Hu, "Direct tunneling leakage current and scalability of alternative gate dielectrics," *Appl. Phys. Lett.*, vol. 81, no. 11, p. 2091, 2002.
- [44] M. Donato, R. I. Bahar, W. R. Patterson, and A. Zaslavsky, "A simulation framework for analyzing transient effects due to thermal noise in sub-threshold circuits," in *Proc. GLSVLSI*, Pittsburgh, PA, USA, 2015, pp. 45–50.
- [45] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*. New York, NY, USA: Springer-Verlag, 2002.
- [46] G. Guennebaud and B. Jacob. (2010). *Eigen v3*. [Online]. Available: <http://eigen.tuxfamily.org>
- [47] Silvaco, "Parallel SmartSpice: Fast and accurate circuit simulation finally available," *Simulat. Stand.*, vol. 8, no. 5, pp. 1–3, 1997.
- [48] A. Hajimiri, S. Limotyrakis, and T. H. Lee, "Jitter and phase noise in ring oscillators," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, pp. 790–804, Jun. 1999.



Marco Donato (S'14–M'17) received the B.S. and M.S. degrees in electrical engineering from the Sapienza University of Rome, Rome, Italy, in 2008 and 2010, respectively, and the Ph.D. degree in electrical sciences and computer engineering from Brown University, Providence, RI, USA, in 2016.

He is currently a Post-Doctoral Fellow and a Lecturer with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA, where he is researching on novel nonvolatile embedded memories with application to machine learning hardware accelerator SoC.



R. Iris Bahar (S'93–M'96–SM'12) received the B.S. and M.S. degrees in computer engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 1986 and 1987, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Colorado at Boulder, Boulder, CO, USA, in 1995.

From 1987 to 1992, she was with Digital Equipment Corporation, Hudson, MA, USA, researching on microprocessor hardware design.

Since 1996, she has been with the School of Engineering, Brown University, Providence, RI, USA, where she is a Professor of Engineering. Her current research interest includes design of energy-efficient and reliable computer systems, from the architecture to device level.

Dr. Bahar was a past Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS.



William R. Patterson (M'73–LM'07) received the B.Sc. degree in physics and the M.Sc. degree in electrical engineering from Brown University, Providence, RI, USA, in 1963 and 1966, respectively.

Since 1977, he has been with the Electrical Sciences Group, Division of Engineering, Brown University, where he is currently a Senior Lecturer and a Senior Research Engineer. His current research interests include low-power analog circuit design for biomedical applications, circuits, and architectures for probabilistic logic, and the instrumentation for geological spectroscopy.

Mr. Patterson was a recipient of the NASA Public Service Medal for his contributions to the Viking Program in 1977.



Alexander Zaslavsky received the B.A. degree in physics from Harvard University, Cambridge, MA, USA, in 1986, and the M.Sc. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA, in 1988 and 1991, respectively.

From 1991 to 1993, he was a Post-Doctoral Scientist with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, researching on SiGe tunneling devices. Since 1994, he has been with the School of Engineering, Brown University, Providence, RI, USA, where he is a Professor of

Engineering and Physics. His current research interest includes semiconductor device and nanostructure physics, particularly tunneling and hot-electron devices, and reliable computing with nanotransistors.

Dr. Zaslavsky has been an Editor of *Solid State Electronics* since 2003.