Towards An Intrinsically Statistical SPICE-Level Simulator

Michel Berkelaar, Qin Tang, Amir Zjajo, Javier Rodríguez, Nick van der Meijs
Circuits and Systems, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
michel@berkelaar.org

***Abstract*—As local random variations on Integrated Circuits are increasingly impacting circuit behavior, there is an increasing need to analyze the impact of these variations efficiently. A widely used tool to analyze circuit behavior is the SPICE-level simulator. To analyze the impact of random variations, these simulators are used inside a CPU-intensive Monte-Carlo loop in current industry practice. In this paper we show that it is possible to build a SPICE-level simulator with built-in statistical capabilities, to improve dramatically on the run time efficiency.**

I. Introduction

Due to advances in semiconductor technology on-chip variations are rapidly increasing in importance. Many of these variations are inherently random, and statistical analysis is needed to assess their impact. For analysis of the behavior of digital and analog circuits at the transistor level typically a SPICE-level simulator is used. To make such an analysis statistical, current industry practice is to add a Monte-Carlo (MC) loop around the simulator. The simulation is repeated many times with different values of the random parameter(s) to sample the statistical behavior of the circuit. To analyze the impact of even just a single statistical parameter, typically thousands of simulations are needed to obtain the statistics accurately. This is due to the poor convergence of MC techniques: $1/\sqrt{n}$, with n the number of samples. If the number of random variables increases, the number of required simulations quickly becomes impractical. Thus, a MC SPICE-level analysis may still work if just the effect of one or two global variations is studied, but local variations are quickly becoming the dominant source of variability in circuit behavior. Unfortunately, the statistical simulation of a circuit with multiple transistors suffering from local variations always involves many random variables, as each transistor is impacted by several local variations in parameters like length (L), width (W), threshold voltage (V_t), etc. It is clear that MC methods (even improved ones using importance sampling) are unsuited for this analysis.

In this paper we explore the possibility of solving the circuit equations of a circuit which depends on random variations directly, eliminating the need for the MC outer loop. This is a hard problem. Normal non-statistical SPICE-level simulation already involves solving a (large) set of time-varying non-linear ordinary differential equations (ODEs). Adding random variables turns this into the problem of solving a set of non-linear random differential equations (RDEs). As a consequence, all the voltages and currents which we calculate will also turn into distributions. This requires the use of new algorithms inside the simulator.

This research was sponsored by the European Community in the ENIAC/MODERN project ENIAC-120003

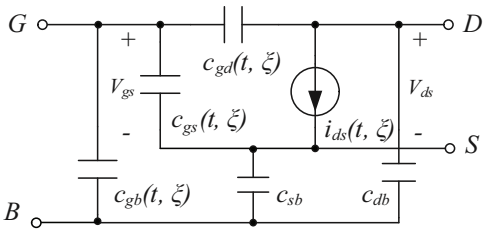


Fig. 1. Transistor Model

In Section III it is shown that efficient algorithms exist which enable the direct solution of the RDEs in a statistical Spice-level simulator. This solution is general in the sense that there are no limitations to the statistical distributions of the random variables, and that correlations are fully accounted for.

Note: this paper explores RDEs only, which limits the applicability to random variables which are *stationary*, ie the statistics of the variables do not change during simulation. This is true for basically all interesting sources of CMOS variability. If we are interested in studying the effects of non-stationary randomness (like noise), we enter the more general domain of stochastic differential equations, which require other, more complex algorithms. Direct solutions for such systems exist as well. See for example [1] for a possible solution to such systems.

II. Statistical Transistor Models

A SPICE-level simulator models the behavior of transistors in so-called companion models. Well-known MOS models are the BSIM and PSP series of models [2]. For a statistical SPICE-level simulator, the transistors need to have statistical models, which means the dependence on the statistical parameters of interest ξ needs to be explicitly available. The simplest possible statistical models use sensitivities (partial derivatives) to obtain the statistics. These sensitivities can either be explicitly available in the model in the form of equations or tables, or obtained by performing a finite difference calculation.

Our simulator is currently written in Matlab, and we do not have access to Matlab versions of the BSIM4 model for the 45nm and 32nm technologies which we use in our experiments. Therefore, in our experiments we use a simplified table-based model [3] targeted at fast simulation of CMOS digital logic, as depicted in Figure 1. There is no technical reason the algorithms explained below could not be used with any existing accurate SPICE transistor model.

III. RDE-based Statistical Simulation

Given a circuit to simulate, we derive the equations which describe its behavior by applying the Kirchhoff current and voltage laws, a process often called Modified Nodal Analysis (MNA). The MNA equation can be written in the compact format:

$$F(\dot{x}, x, t, p_0) = 0 \qquad x(t_0) = x_0 \tag{1}$$

where x denotes the state variable vector including node voltages, $\dot{x}$ is its time derivative and p_0 represents the nominal process parameter value vector.

Denote $x_s(t)$ as the solution of (1) which satisfies:

$$F_s = F(\dot{x_s}, x_s, t, p_0) = 0 \qquad x(t_0) = x_0 \tag{2}$$

Since all process parameters have their nominal values p_0, $x_s(t)$ is deterministic, which means it can be solved by well-known methods as employed in the current SPICE-level simulators. If we take into account process variations, (1) becomes a random differential equation (RDE):

$$F_x = F(\dot{x}, x, t, \xi) = 0 \qquad x(t_0) = x_0 + \delta_0 \tag{3}$$

where ξ is the process variation vector which includes both global and local variations, and δ_0 denotes the initial condition variation caused by process variations. It is worth noticing that the main difficulty to solve (3) lies in the high nonlinearity with respect to the random variables ξ and the large number of process variations including local variations. In order to make (3) manageable, it is linearized by a truncated Taylor expansion around x_s and p_0.

$$\begin{aligned} F_x \;\approx\; & F_s + \frac{\partial F_s}{\partial \dot{x_s}}(t)(\dot{x}(t) - \dot{x_s}(t)) \\ + \; & \frac{\partial F_s}{\partial x_s}(t)(x(t) - x_s(t)) + \frac{\partial F_s}{\partial p_0}(t)\xi = 0 \end{aligned} \tag{4}$$

To simplify the notation, the variation of state variable x is denoted by y, thus $x(t)$ can be rewritten as $x(t) = x_s(t) + y(t)$. Inserting this to (4) and replacing the matrices $\partial F_s/\partial \dot{x_s}$, $\partial F_s/\partial x_s$ and $\partial F_s/\partial p_0$ with $\mathbf{C(x_s)}$, $-\mathbf{E(x_s)}$ and $-\mathbf{F(x_s)}$, respectively, we obtain:

$$\mathbf{C}(x_s)\dot{y}(t) = \mathbf{E}(x_s)y(t) + \mathbf{F}(x_s)\xi \qquad y(t_0) = y_0 = \delta_0 \tag{5}$$

$\mathbf{C}$, $\mathbf{E}$ and $\mathbf{F}$ are $N_v \times N_v$, $N_v \times N_v$ and $N_v \times N_p$ matrices respectively, where N_v is the number of unknown nodes and N_p is the number of process variations. Consequently, the nonlinear equation (3) is converted to a linear RDE in y with x_s-dependent coefficient matrices.

$x_s(t)$ can be solved by well-known deterministic methods like in any SPICE-level simulator.

Unfortunately, the variation of state variable $y(t)$ can not be calculated directly from (5) since ξ is a random variable. According to the Random Differential Equation (RDE) theorem [4], (5) has a unique mean square solution which can be represented by:

$$y(t) = \Phi(t, t_0)y_0 + \Theta(t)\xi = \mathbf{\Psi}(t)\xi \tag{6}$$

where $\Phi(t, t_0)$ is the homogeneous solution of (5) satisfying

$$\mathbf{C}(x_s)\dot{\Phi}(t, t_0)) = \mathbf{E}(x_s)\Phi(t, t_0) \tag{7}$$

and $\Theta(t)$ is an integral in the range $[t_0, t]$, which depends on Φ, $\mathbf{C}$ and each column of $\mathbf{F}$ [5]. If the initial condition x_0 is deterministic, then y_0 is zero. For some types of simulations, the voltage variation can be considered zero at time t_0. Then, the initial condition for our problem is deterministic. Even if the initial condition y_0 is statistical due to process variations, it can also be represented as a first-order function w.r.t. ξ. Therefore, $y(t)$ can be rewritten as $\mathbf{\Psi}(t)\xi$ in (6) where $\mathbf{\Psi}(t)$ is a $N_v \times N_p$ matrix.

We obtain $\mathbf{\Psi}(t)$ by substituting (6) into (5):

$$\mathbf{C}(x_s)\dot{\mathbf{\Psi}}(t) = \mathbf{E}(x_s)\mathbf{\Psi}(t) + \mathbf{F}(x_s) \tag{8}$$

After solving x_s and $\mathbf{\Psi}(t)$, $x(t)$ can be obtained based on $x(t) = x_s(t) + y(t)$ and $y = \mathbf{\Psi}(t)\xi$ in (6).

$$x(t) = x_s(t) + \mathbf{\Psi}(t)\xi \tag{9}$$

Equation (9) is used to calculate the time-varying moments of voltages. The first two central moments and covariance are expressed in (10)-(12), where the correlation coefficients ρ between every pair of process variations are included in the $E\{\xi\xi^T\}$ calculation.

$$E\{x(t)\} = x_s(t) \tag{10}$$

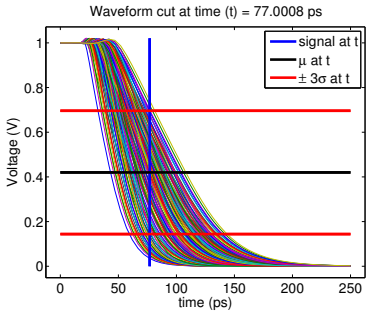
$$Var\{x(t)\} = \mathbf{\Psi}(t)E\{\xi\xi^T\}\mathbf{\Psi}^T(t) \tag{11}$$

$$Cov\{x(t_a), x(t_b)\} = \mathbf{\Psi}(t_a)E\{\xi\xi^T\}\mathbf{\Psi}^T(t_b) \tag{12}$$

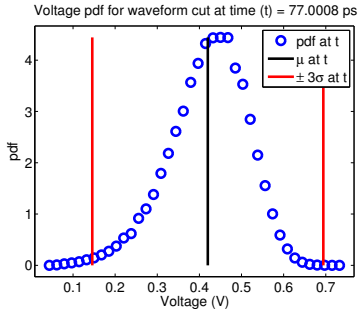
As becomes clear from the above, in a statistical simulator voltages and currents are no longer simple values but distributions. In a SPICE-like simulator this means that for every time t, the statistical distributions of voltages and currents are obtained. Figure 2 shows a variable waveform, and how at every time t a distribution is associated with the voltage.

IV. From Voltage To Delay Distribution

In the results section below we will analyze some digital circuits. In digital circuits we are not just interested in voltage waveforms, but also in the notions of arrival time and delay. These are related to the time of crossing a certain reference voltage, often the 50% of supply voltage level. The crossing time t_η is defined as the first time for voltages to cross the threshold voltage $V_\eta = \eta\% \cdot V_{dd}$. The *cdf* of crossing time is calculated when the nominal voltage is in transition. For a



(a) Statistical Waveforms



(b) Probability of Voltage

Fig. 2. Statistical Voltage

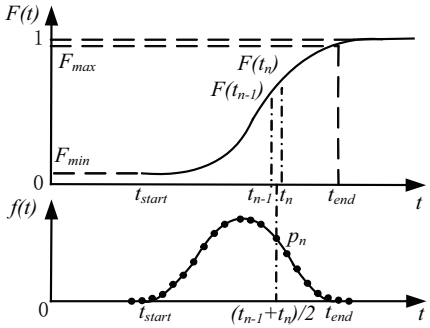


Fig. 3. Cumulative distribution function (*cdf*) and discretized probability density (*pdf*) function

rising transition this is expressed as:

$$F_n = P(t_\eta \le t_n) = 1 - P(t_\eta > t_n) = 1 - G_n \quad (13)$$

$$G_n = P(v_1 \le V_\eta \cap v_2 \le V_\eta \cap \cdots \cap v_n \le V_\eta) \quad (14)$$

$$= P(v_n \le V_\eta | v_{n-1} \le V_\eta, \ldots, v_1 \le V_\eta) \cdot G_{n-1} \quad (15)$$

$$= P(v_n \le V_\eta | v_{n-1} \le V_\eta) \cdot G_{n-1} (n = 2 : N) \quad (16)$$

$$= \frac{P(v_n \le V_\eta \cap v_{n-1} \le V_\eta)}{P(v_{n-1} \le V_\eta))} \cdot G_{n-1} \quad (17)$$

where v_i is the voltage of interest at time t_i and F_n denotes the *cdf* of crossing time at time t_n. Equation (15) is rewritten in (16) since the voltages are modeled as Markovian processes [6], [7]. Based on (13) to (17) an iteration method is used to calculate the *cdf* of the corresponding crossing time with initial condition G_1=1. Given the moments and covariances calculated in the RDE-based statistical simulator in (10)-(12), the joint probability and single probability in (17) can be obtained.

The relationship between the *cdf* and the discretized *pdf* (denoted by F and f respectively) in our algorithm is illustrated in Fig. 3. If the simulation uses a non-uniform time step algorithm, the *cdf* needs to be uniformly sampled for *pdf* computation. After uniformly sampling and interpolating from the effective *cdf* with N_s samples, the $N_s \times 1$ time and *cdf* vectors are obtained and denoted as T_1 and cdf_u, respectively. These vectors are used to calculate the *pdf* vector Ω with element $\Omega_k = cdf\,u_k - cdf\,u_{k-1}$ ($\Omega_1 = 0$, $k = 2 : N_s$).

The last step is to calculate the moments of crossing time in which we are interested. As an example, mean μ, standard deviation σ and skewness γ are used. Denoting T_1^T as the transposition of the column vector T_1, the calculation method can be formulated as follows [7]:

$$\mu = T_1^T \Omega \qquad \sigma = T_2^T \Omega - \mu^2 \quad (18)$$

$$\gamma = (\Gamma - 3\mu\sigma^2 - \mu^3)/(\sigma^3) \quad (\Gamma = T_3 \Omega^T) \quad (19)$$

The relationships between the elements of T_2 and T_3 with T_1 are $T_2(k) = T_1^2(k)$ and $T_3(k) = T_1^3(k)$ where $k = 1 : N_s$.

The calculation method for a falling transition is similar to the above methods with the only difference in (14) where v_i is replaced by $V_{dd} - v_i$. If the waveform is non-monotonic and crosses V_η multiple times, the method above can be used to iteratively find all crossing times.

V. Results

A. Variability in Digital Combinational Logic

We can simulate a digital cell and obtain the output voltage statistics. From these voltage statistics we can obtain the statistical distribution of the 50% crossing time (see Figure 4). From this we can calculate the moments (mean, standard deviation, higher order moments if we want) of the delay we are interested in [8]. The result we obtain in this way could also be obtained by Statistical Static Timing Algorithms (SSTA), but please observe that in our solution we can handle arbitrary statistical distributions of the process parameters, and handle the statistical correlation between the signals in the circuit, while maintaining SPICE-level accuracy. As [8] shows, our method is very accurate, with mean errors less than 1% and σ errors less than 6% for a 45nm technology library [9], even with strong correlations between input signals. In these experiments we used 2 process parameters which vary, the transistor length L and the threshold voltage V_t. The basis for this comparison are extensive ($10,000$ iterations) Monte Carlo (MC) simulations in Cadence Spectre with a BSIM4 transistor model for the technology. The run time overhead of our statistical extensions is very small, especially when compared to MC iterations; adding L and V_t as random variables increases the simulation run time of our MatLab code by only 40%.

B. Variability in a D-Flipflop

In the analysis of digital logic, we are not restricted to feedback-free combinational logic. As an example we look at a D-flipflop from the same 45nm library, see Figure 5. With

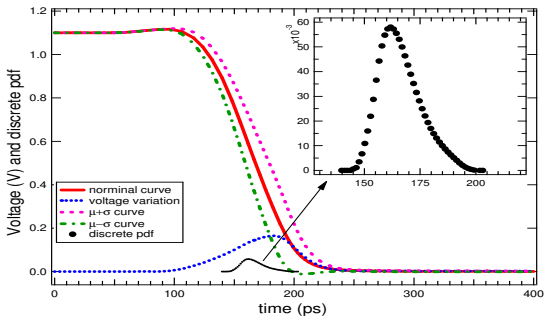


Fig. 4. Variational waveforms of AOI211 and its discrete *pdf*. The discrete *pdf* of 50% crossing time is shown enlarged on the upper right corner.

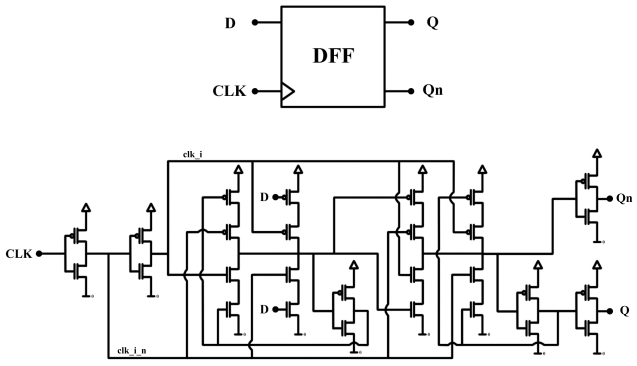


Fig. 5. D-FlipFlop schematics

28 transistors and multiple feedback paths this is already a non-trivial example. Figure 6 shows the distributions of the clock input CLK to the Q and Q_n outputs. When compared to again MC Spectre runs with a BSIM4 model, means errors are again smaller than 1%, and σ errors smaller than 5%, for a range of inputs signals and output loads. The run time in our Matlab simulator is about 50 seconds, the MC Spectre run takes about 40 minutes. This shows that even though we have a slow Matlab implementation, we are already considerably faster than a highly optimized industrial tool for this task.

C. Variability in Digital Sequential Logic

We can now also look at a very simple digital sequential circuit, where the launching and catching flipflops are fully included. In Figure 7 the circuit with 3 flipflops and an AND-gate is depicted. The circuit includes wire resistance

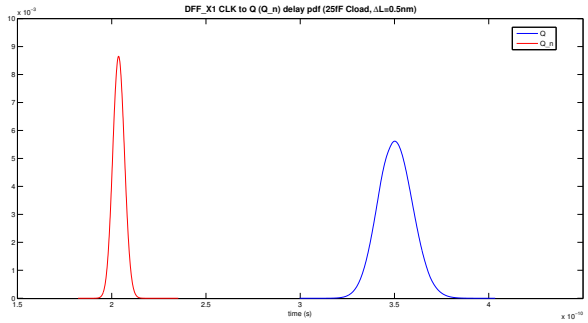


Fig. 6. D-Flipflop delay

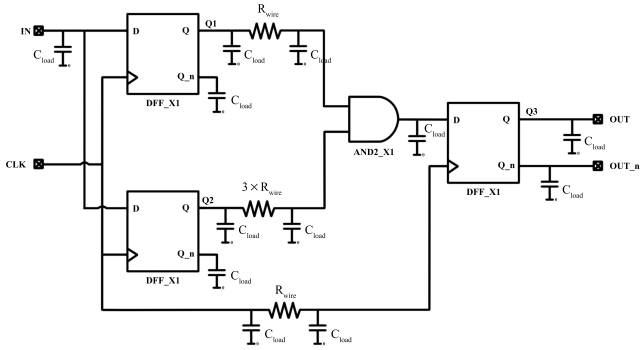


Fig. 7. Simple sequential circuit

and capacitance, both for the signal as well as for the clock nets. Note that this circuit has a non-zero clock skew. For this circuit, the maximum error of mean delay at the outputs of all three flipflops is again smaller than 1%, the error in σ is smaller than 8%. The run time for our prototype Matlab simulator is a few minutes, the MC iterations in Spectre take a few hours.

VI. Conclusion

This paper shows that methods do exist to perform statistical SPICE-level simulation efficiently without the use of Monte Carlo iterations. This leads to a huge run time improvement. In this paper digital circuit behavior is analyzed, both combinational as well as sequential. In the future it will be interesting to study more general analog circuit behavior impacted by parameter variation.

References

[1] A. Zjajo, Q. Tang, M. Berkelaar, J. P. de Gyvez, A. D. Bucchianico, and N. van der meijs, "Stochastic analysis of deep-submicrometer cmos process for relaible circuits designs," *IEEE Transactions on Circuits and Systems*, vol. 58, no. 1, pp. 164–175, 2011.

[2] BSIM4. [Online]. Available: http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html

[3] Q. Tang, A. Zjajo, M. Berkelaar, and N. P. van der Meijs, "Transistor level waveform evaluation for timing analysis," in *Proceedings of European Workshop on CMOS Variability (VARI)*, 2010, pp. 1–6.

[4] T. T. Soong, *Random differential equations in science and engineering*. New York: Academic Press, 1973.

[5] Q. Tang, A. Zjajo, M. Berkelaar, and N. P. van der Meijs, "RDE-based transistor-level gate simulation for statistical static timing analysis," in *Proceedings of IEEE Design Automation Conference*, 2010, pp. 787–792.

[6] H. Fatemi, S. Nazarian, and M. Pedram, "Statistical logic cell delay analysis using a current-based model," in *Proceedings of IEEE Design Automation Conference*, 2006, pp. 253–256.

[7] Q. Tang, A. Zjajo, M. Berkelaar, and N. P. van der Meijs, "Statistical delay calculation with multiple input simultaneous switching," in *Proceedings of IEEE International Conference on IC Design and Technology*, 2011, pp. 1–4.

[8] ——, "Transistor-level gate model based statistical timing analysis considering correlations," in *Proceedings of IEEE Design, Automation and Test in Europe Conference*, 2012, pp. 917 – 922.

[9] Nangate 45nm open cell library 2009. [Online]. Available: http://www.nangate.com/index.php?option=com_content&task=view&id=137&Itemid=137