This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A Fast Modular Method for True Variation-Aware Separatrix Tracing in Nanoscaled SRAMs

Adam Teman, *Member, IEEE*, and Roman Visotsky

***Abstract*— As memory density continues to grow in modern systems, accurate analysis of static RAM (SRAM) stability is increasingly important to ensure high yields. Traditional static noise margin metrics fail to capture the dynamic characteristics of SRAM behavior, leading to expensive over design and disastrous under design. One of the central components of more accurate dynamic stability analysis is the separatrix; however, its straightforward extraction is extremely time-consuming, and efficient methods are either nonaccurate or extremely difficult to implement. In this paper, we propose a novel algorithm for fast separatrix tracing of any given SRAM topology, designed with industry standard transistor models in nanoscaled technologies. The proposed algorithm is applied to both standard 6T SRAM bitcells, as well as previously proposed alternative subthreshold bitcells, providing up to three orders-of-magnitude speedup, as compared with brute force methods. In addition, for the first time, statistical Monte Carlo separatrix distributions are plotted.**

***Index Terms*— Control theory, dynamic noise margin (DNM), Monte Carlo (MC) simulation, phase portrait, separatrix, SRAM, stability analysis, static noise margin (SNM).**

I. INTRODUCTION

AS THE predominant embedded memory technology in modern ICs, static RAM (SRAM) has become the primary consumer of both power and silicon area in nanoscaled IC systems. To integrate the largest possible amount of SRAM, the traditional six-transistor (6T) SRAM bitcell has closely followed the aggressive scaling of CMOS technologies, and has employed nonstandard layouts (pushed rules) to achieve even higher densities. These trends, which according to the latest International Technology Roadmap for Semiconductors update [1], are expected to continue for at least another decade, and are accompanied by an unavoidable increase in process variations that lead to a loss of data stability [2]. Supply voltage (V_{DD}) scaling, targeted at lowering the power consumption of both core logic and SRAM blocks, further impedes the robustness of SRAM bitcells due to reduced noise margins and increased sensitivity to device parameter fluctuations. Furthermore, the high density and chip area consumption of SRAM blocks makes them susceptible to soft errors caused by external radiation as well as other single-event-upsets (SEUs), such as those caused by coupled signals, further impairing data integrity [3]. Data stability has always been a central issue in SRAM design, but due to the aforementioned trends, traditional static noise margin (SNM) metrics for ensuring this stability have become impossible to maintain along with the aggressive scaling and density aspirations, and are insufficient for measuring durability in high radiation environments or due to SEUs [2]–[6].

The importance of dynamic noise modeling and stability for both logic and SRAM circuits was first noted over three decades ago [7]. However, dynamic analysis unequivocally took a back seat to SNM criteria for SRAM stability evaluation. Only recently, due to the increasing challenges in achieving high SRAM yields in modern systems, dynamic noise margins (DNMs) came into focus [2]–[5], [8]–[17]. A primary challenge to the widespread adoption of dynamic criteria is the ability to characterize dynamic stability in a simple, straightforward fashion, such as the single number SNM metric [6]. However, as opposed to static margins, which are modeled with a constant dc serial noise source, dynamic noise requires, at the minimum, a 2-D noise source, characterized by both amplitude and duration [3]. Therefore, despite several attempts to define a single number metric for DNM [2]–[5], [10]–[12], a toolbox of graphs and metrics is still required to evaluate stability [10]. Of these, the most important is probably the well-known separatrix [9], [11], [13].

In this paper, we present a novel algorithm for fast and accurate separatrix tracing. While the majority of the previously proposed methods [2], [5], [9], [11], [13] can be applied only to a specifically and rigorously modeled 6T bitcell, rely on nonaccurate transistor models, require a custom solver, cannot take into consideration statistical device variations, and/or are complex to implement, the proposed algorithm:

1) is modular and topology independent, i.e., it can be applied to any given bistable circuit;
2) uses technology provided device models (such as the 40-nm BSIM4 models used for demonstration), providing the highest available accuracy;
3) is operated with any industry standard SPICE solver, such as Cadence Spectre, used in this paper;
4) can consider global and local statistical variations, as provided with the technology models;
5) is configurable to conveniently tradeoff accuracy versus runtime, as required;
6) is simple to comprehend, implement, and apply, providing the designer with an immediate solution for separatrix tracing, and therefore DNM analysis.

The proposed algorithm is shown to provide a 1000× speedup over brute-force separatrix tracing with the same

Manuscript received April 2, 2014; revised August 12, 2014; accepted September 9, 2014.

A. Teman is with the Telecommunications Circuits Laboratory, Institute of Electrical Engineering, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: adam.teman@epfl.ch).

R. Visotsky is with Dolphin Integration, Netanya 42504, Israel (e-mail: viso.rom@gmail.com).

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/TVLSI.2014.2358699

1063-8210 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

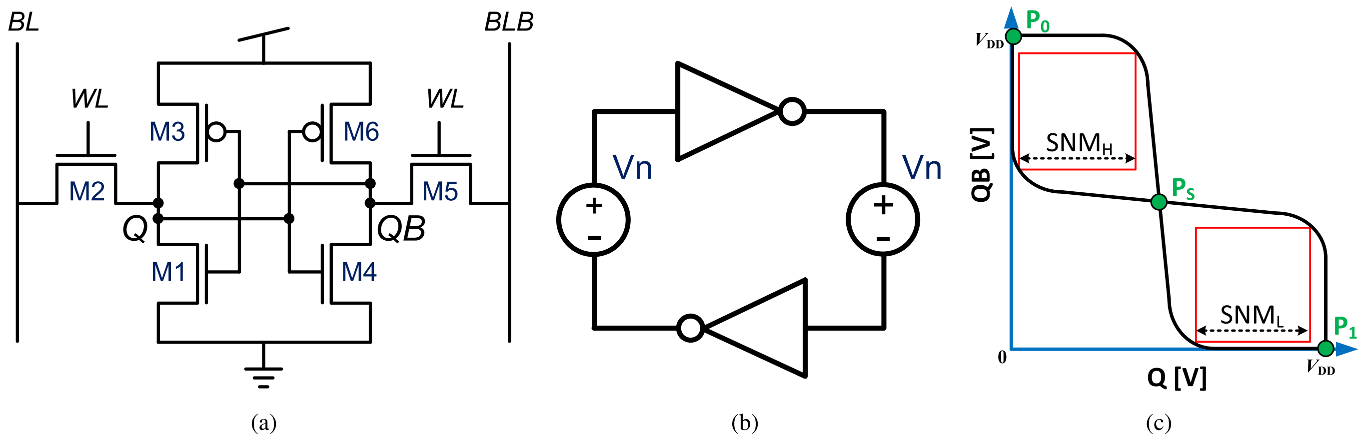


Fig. 1. (a) Schematic of a 6T SRAM bitcell. (b) Conceptual definition of SNM. (c) Maximum square method for SNM calculation, demonstrated upon a hold butterfly curve plot.

resolution. Consequently, these reduced run times allow application of advanced and complex analysis, previously unavailable, such as statistical Monte Carlo (MC) simulation. In this paper, for the first time, statistical separatrix distributions for SRAM bitcells are displayed. Furthermore, the proposed algorithm is used to trace the separatrix of various nonstandard SRAM bitcell circuits, designed for specific applications, such as subthreshold operation [17]–[19].

The rest of this paper is organized as follows. Section II provides an overview of SRAM stability analysis from both a static and dynamic perspective, including its traditional limitations. Section III presents the theory proving the existence of the separatrix and leading to the development of the proposed algorithm. The proposed algorithm is presented in Section IV and shown to correctly trace separatrices under statistical variations in Section V. Section VI presents the separatrices of several alternative SRAM topologies, as traced with the proposed algorithm. Section VII concludes this paper.

II. SRAM Stability Analysis

Since the widespread adoption of CMOS process technology, the most common circuit for implementing single-ported SRAM cells is the 6T bitcell, shown in Fig. 1(a). The core of this cell is made up of a pair of cross-coupled CMOS inverters (M1/M3 and M4/M6) that form a bistable circuit with the stored data represented by the voltages of the complementary internal data nodes (Q and QB). The strong positive feedback between the two inverters results in robust bistability, enabling high yields and noise immunity, which have led to the popularity of this topology. Most other SRAM solutions, such as multiported bitcells and bitcells targeted at subthreshold operation, are also based on this internal core. Therefore, all discussions of SRAM stability start, and often end, with analysis of the 6T cell.

Initial discussions of digital noise margins were intended for logic gates [7], [20]–[22], primarily focusing on the dc operating points of these circuits, as extracted from their voltage transfer characteristics (VTC). With the 6T SRAM core made up of a pair of logic gates, these discussions were famously adapted for SRAM in [6]. Seevinck *et al*. [6] defined the SNM of an SRAM cell as the largest serial voltage noise that can be oppositely applied to the complementary data nodes of the 6T circuit without losing bistability. The conceptual setup of this definition is shown in Fig. 1(b), with V_n representing the magnitude of the DC noise sources. This definition had previously been shown to be mathematically equivalent to several other popular noise margin definitions for logic gates [21]. What caused the breakthrough was Seevinck's description of a fast simple SPICE compatible method for measuring one of these equivalent definitions—the *maximum square method*.

The maximum square method, shown in Fig. 1(c), defines digital noise margins as the side of the largest square that can fit into the lobes of the butterfly curve plot of a logic gate. Butterfly curves are obtained by plotting the VTCs of odd and even stages of a chain of inverting gates on a single plot. As the cross-coupled inverter structure of the 6T SRAM essentially creates an infinite chain of inverting gates, this plot appropriately shows the dc characteristics of the bitcell. The logic levels of the SRAM cell are represented by the two stable operating points of this graph [the intersecting points of the VTCs, marked as P_0 and P_1 in Fig. 1(c)], with the middle intersection (marked as P_S) a metastable point. Inserting serial dc voltage sources of opposite polarities at Q and QB causes the VTCs to move toward each other, depleting the lobes of the butterfly curves. If a voltage larger than the side of the maximum square is applied, the VTCs cease to intersect at three points, resulting in a loss of bistability. Seevinck showed that by applying a rotation transformation to the butterfly curves and subsequently subtracting the two VTCs, the diagonal length of the squares that fit into the lobes is plotted. This enables a simple extraction of the maximum square, using a single dc sweep and a rotation circuit that can be implemented in any standard circuit simulator.

One of the most important characteristics of the Seevinck method for measuring SRAM stability is its compatibility with statistic parametric analysis. The SNM can be measured under MC statistical sampling to provide stability distributions of SRAM cells in light of global process variations and local

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

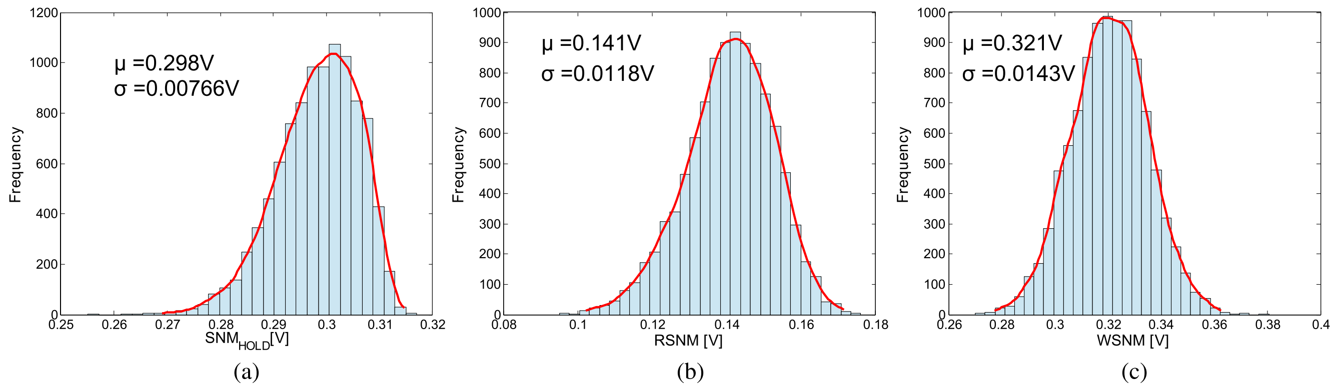


Fig. 2. SNM Distributions for a 40-nm 6T bitcell with $V_{\mathrm{DD}} = 1$ V (10k MC samples of global and local variations at 25 °C). (a) Hold SNM. (b) RSNM. (c) WSNM.

mismatch. An example distribution is shown in Fig. 2(a) for a 40-nm 6T bitcell, biased at its nominal 1-V supply voltage. This distribution shows that not only is the mean (μ) SNM relatively high (in this case 298 mV), but it is nonnegative over many standard variations (σ). This analysis is used to ensure high yield for millions of products, each containing millions of bitcells. In addition, by simply changing the node biases in the testbench, read and write operations can be simulated, providing similar distributions for read SNM (RSNM) and write SNM (WSNM), as shown in Fig. 2(b) and (c), respectively, for the same circuit. These metrics show depleted margins in terms of both mean and standard deviation as compared with hold SNM, appropriately tracking the increased sensitivity of the 6T circuit during access operations. In addition, several alternative metrics for WSNM and RSNM have been considered with various advantages and drawbacks as compared with the Seevinck method [12], [23]–[25].

Despite being the defacto standard for SRAM stability, the SNM metric has many flaws. As a static metric, it fails to capture the fundamental nonlinear dynamics of the 6T bitcell as well as the time-dependent characteristics of read, write, and SEU events. It assumes infinite dc biases in a standalone environment, rendering it unable to capture the effects of the array architecture, peripheral circuitry, and power supplies. Finally, and maybe most significantly, the SNM metric models a nonphysical noise source—a serial voltage source with opposite polarities at the data nodes, which on the one hand represents a much more extreme situation that can actually occur, and on the other hand fails to provide a metric that can be captured in terms of an actual noise specification. As a result of these flaws, designing SRAMs exclusively based on the SNM metric leads to overdesign in some cases, and underdesign in others. For hold analysis, the extreme noise modeling leads to overdesign, requiring unnecessary margins that are increasingly hard to achieve in state-of-the-art technologies and under aggressive layouts. Read operations assume infinite access time, causing artificial failures when a state flip would only occur after the deassertion of the word line, while not considering the time-dependent discharge of the bitlines or any coupling effects during signal transistions. WSNM analysis, while also disregarding the dynamic behavior of the system, can further lead to missed write failures due to the assumed infinite write pulse, as the actual access time may be too short to cause a state flip.

Accordingly, several groups have started to focus on developing methods to evaluate the dynamic stability of SRAMs and define DNMs [2]–[5], [9]–[12], [16]. Dynamic noise analysis most often introduces noise as a time-variant current pulse applied to one or more circuit nodes. This enables modeling physical noise sources, such as particle strikes and coupling. In addition, actual access operations can be simulated by applying read and write operations to the full array architecture. Finally, alternative SRAM implementations, which rely on various dynamic operations or behavior during access or state variation, can be correctly modeled for an impartial comparison.

Whereas dynamic stability analysis provides many levels of improvement and accuracy over static metrics, it also adds inherent complexity. Evaluating the stability of an SRAM topology with a static metric is straightforward, based on the μ and σ of its SNM distribution. However, dynamic stability lacks a universally accepted definition, and furthermore depends on many varying factors for analysis, such as the duration of access pulses and the exact attributes of a noise event. The most well known definitions of SRAM dynamic stability are those described in [26] and [27], providing a failure criterion for an SRAM array under noise or an access operation. While these definitions provide a means to estimate the yield of a given design, operated at a specific voltage, speed, and so on, they do little to provide insight into the actual causes of failure, and therefore are ineffective as a tool for improving a design. The additional information provided by DNM measurements can fill this gap.

Several groups have provided definitions for DNM; however, none of these has yet to emerge as a clear cut winner. Lostroh [7] introduced the noise immunity curve, presenting a shmoo plot showing regions of failure and success for a logic gate in the presence of a trapezoidal noise pulse of a given amplitude and duration. Zurada *et al.* [28] displayed dynamic (ac) VTCs for logic gates, which led to Ding and Mazumder's [4] definition of maximum square-based DNMs. However, these definitions do not sufficiently capture the dynamic requirements of SRAM stability, leading to the definition of DNM by Dong *et al*. [5], as the margin of time between

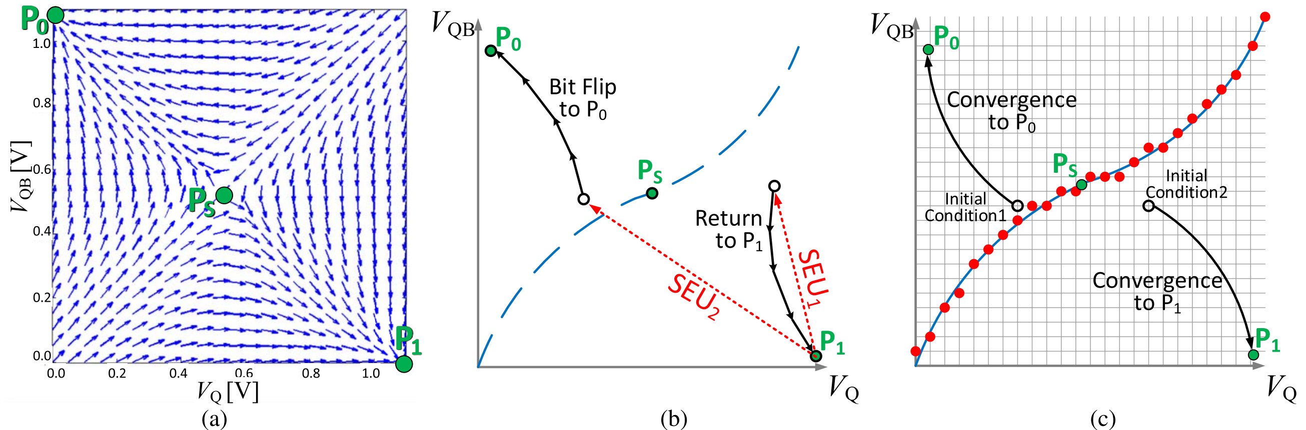


Fig. 3. (a) Phase portrait of a 6T SRAM cell. (b) Demonstration of reconvergence or bit-flip due to an SEU. (c) Brute force state-space sampling.

the applied pulse and the minimum time to cause a state flip (in write) or maintain data retention (in read). Wang *et al.* [12] propose a similar model for write DNM, modeling the injected current waveforms during a write access and measuring the critical time for a successful write operation. Sharifkhani and Sachdev [10] invoke small signal loop-gain analysis to show the existence of stable states and rigorous state-space analysis to test dynamic stability criteria. Wieckowski *et al.* [11] define *separatrix affinity* as a metric for measuring the dynamic stability of SRAM cells, including nonstandard topologies, such as their portless 5T cell.

As a result of the wide array of, often contradicting, definitions for DNM, we have reached the following conclusions.

1) To carry out dynamic stability analysis, no matter what the chosen DNM definition is, a toolbox of graphs and metrics is required.
2) The most important of these tools is the stability boundary between the regions of attraction of the stable states. This boundary, better known as the *separatrix*, is defined, presented, and discussed in the next section.

III. Separatrix Theory

The 6T SRAM bitcell is a dynamic nonlinear bistable system requiring rigorous nonlinear control theory for mathematical analysis. The behavior of the circuit can be described by choosing the internal data node voltages [V_Q and V_{QB} from Fig. 1(a)] as the state variables and writing the state equations [9]

$$\begin{cases} \partial V_{\mathrm{QB}}(t)/\partial(t) = f_f(V_Q, V_{\mathrm{QB}}) \\ \partial V_Q(t)/\partial(t) = f_b(V_Q, V_{\mathrm{QB}}) \end{cases} \tag{1}$$

where $f_f(\cdot)$ is the feed-forward nonlinear function from Q to QB and $f_b(\cdot)$ is the nonlinear feedback function from QB to Q. Bistability of this state space can easily be demonstrated with the butterfly curve plot of Fig. 1(c). As previously mentioned, these curves plot the dc characteristics of the cross-coupled inverters that comprise the circuit. A corollary to this is that these curves also trace the points at which the derivatives of f_f and f_b are zero, better known as the *nullcline vectors* [11]. Therefore, the intersections of the nullclines are points at which

$$\begin{cases} \partial V_{\mathrm{QB}}(t)/\partial(t) = f_f(v_q, v_{qb}) = 0 \\ \partial V_Q(t)/\partial(t) = f_b(v_q, v_{qb}) = 0 \end{cases} \tag{2}$$

such that the points (v_q, v_{qb}) are equilibria. Therefore, the SRAM circuit has three equilibria, denoted as P_0, P_1, and P_S.

Evaluation of the stability of a particular equilibrium point is achieved by analyzing the eigenvalues of the Jacobian matrix around the equilibrium point [29]. For the two extreme points, P_0 and P_1, the real part of the Jacobian eigenvalues are both negative, resulting in stable states, namely, the bitcell's *data* 0 and *data* 1 states, respectively (designated according to the logic state of node Q). The eigenvalues of the third equilibrium P_S on the other hand, have one positive and one negative real parts, and therefore, this is an unstable equilibrium, better known as a *saddle* point.

Plotting the phase portrait of the 6T circuit provides more insight into these conclusions. The phase portrait of a balanced 6T bitcell is shown in Fig. 3(a), with the three equilibria pointed out. Following the phase vectors in the top left of the plot will bring the state space to P_0, while following the bottom right vectors will proceed to P_1. The boundary between the regions of convergence (RoC) or *basins of attraction* of P_0 and P_1 is an unstable manifold made up of phase vectors that lead to P_S. Therefore, a theoretical initial condition upon this manifold would result in dc convergence at the unstable equilibrium; however, any small perturbation from this manifold will result in convergence at one of the stable points. This stability boundary is better known as the *separatrix* of the system [29].

The importance of the separatrix in dynamic stability analysis is straightforward. Given an initial state, for example, P_1 following a *write* 1 operation, an SEU can be simulated. Depending on the amplitude, duration, shape, and attack point of the SEU, the state space of the system will be pushed to a different point on the graph. Following the passing of this noise event, the circuit will either converge back to P_1 or flip and converge to P_0, as shown in Fig. 3(b). The condition for state retention or loss of data is if the state space following the event crossed the separatrix or not. In a similar fashion, both

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
COMPARISON BETWEEN SEPARATRIX TRACING ALGORITHMS

Method	# Sims	Accuracy	Solver	Model	Topology
Brute Force	N^2 transients	High	Standard	Any	Any
Zhang [3]	Analytical solution	Low	None	Simple	Only 6T
Huang [5], [9], [13]	2 DC Sweeps, 2 transients	Low	Custom	Custom	Only specific
Wieckowski [11]	N^2 DC Ops	Medium to High	Standard	Any	Only 2D lumped cap
This work	$\mathcal{O}(NlogN)$ transients	High	Standard	Any	Any

read and *write* operations can be modeled as a dynamic noise event, and knowledge of the separatrix enables evaluation of the success of the operation (i.e., retaining or toggling the stored data). Therefore, the majority of the DNM metrics, mentioned in Section II, rely on knowledge of the separatrix for calculation.

The basic method for separatrix tracing is through brute-force state space sampling. In this method, shown in Fig. 3(c), the state space is divided into a dense grid, and a transient simulation is run with initial conditions for each grid coordinate. The convergence state of each simulation is evaluated, and accordingly, the RoC of each stable point is mapped, revealing the separatrix as the boundary between the two RoCs. This method has often been shown to be computationally inefficient, with a quadratic relationship between run time and accuracy [9]. This all but rules out statistical analysis, a mandatory requirement for high-yield SRAM design.

Accordingly, several methods for the derivation of the separatrix have been proposed in the recent past. Zhang *et al.* [3] developed a closed-form expression for dynamic stability based on nonlinear state space analysis. However, this analysis was based on piece-wise linear device models and assumed the separatrix was a straight line, diagonally separating the state space along the $V_Q = V_{\text{QB}}$ vector. Dong *et al.* [5] and Huang *et al.* [9] proposed an algorithm for separatrix tracing, based on only two transient simulations. They chose an initial point on the separatrix (such as the nonstable equilibrium) and integrated backward to reveal the entire manifold. This analysis introduces variability into the SRAM model, displaying the effect of device mismatch on the separatrix. However, their algorithm requires the extraction of a modified dynamic system that allows the backward integration. This is a nontrivial operation and may not be adaptable to high-complexity device models, such as BSIM4. In [2], the separatrix is approximated as two line segments based on dynamic analysis and alpha-power law device models. Finally, in [11], a black box method for stability analysis is developed with a method for separatrix tracing without the need for any transient analyses or dc convergence by representing each initial condition in the state-space grid with a circuit model.

While these works propose very efficient methods for separatrix tracing, each with its specific advantages (summarized in Table I for the equivalent of $N \times N$ brute-force accuracy), none of them provides a method as simple and modular as the brute-force sampling method but with short enough run times to enable statistical stability analysis. The following section presents our algorithm for separatrix tracing, providing a uniform solution for all of these requirements.

Algorithm 1 Modular Separatrix Tracing

Require: Define process technology, bitcell netlist, V_{DD}, resolution (x_{step}), and tolerance (d_{tol})
1: Guess initial point, $p_0^g = (V_{\text{DD}}/2, y_0^g)$
2: Find point $p_0^u = (V_{\text{DD}}/2, y_0^u)$ on separatrix according to **Algorithm 2**.
3: **for** $i = 1; i \le V_{\text{DD}}/2x_{step}; i = i + 1$ **do**
4: Set $x_- = V_{\text{DD}}/2 - i \cdot x_{step}$; $x_+ = V_{\text{DD}}/2 + i \cdot x_{step}$
5: Guess: $p_-^g = (x_-^g, y_-^g); p_+^g = (x_+^g, y_+^g)$, where y_-^g and y_+^g are set according to a linear approximation of previous points on the separatrix.
6: Find points $p_-^u = (x_-, y_-^u)$ and $p_+^u = (x_+, y_+^u)$ on separatrix according to **Algorithm 2**.
7: Add points p_-^u and p_+^u to Separatrix trace.
8: **end for**

IV. PROPOSED ALGORITHM

The proposed algorithm is presented below in two parts. Algorithm 1 provides the frame procedure that controls the simulation flow, while Algorithm 2 describes the subroutine for finding the separatrix in proximity to a given point. As inputs, the procedure accepts the netlist of the bitcell, composed of device models from any technology, in addition to the target supply voltage and two configuration parameters—the resolution (x_{step}) and the error tolerance (d_{tol}) of the output.

An initial guess is taken for the first point on the separatrix. This point can have any value, but for symmetry, we show the initial V_Q value to be at the middle of the voltage swing ($V_{\text{DD}}/2$), and lacking additional knowledge about the cell, the arbitrary point $p_0^g = (x_0, y_0^g) = (V_{\text{DD}}/2, V_{\text{DD}}/2)$ could be used, where p denotes a point in the state space with coordinates $(x_i, y_i) = (V_Q, V_{\text{QB}})$ and the superscript ^{g} denotes a guess. Using this initial guess, a point with $p_0^u = (x_0, y_0^u)$ within d_{tol} of the separatrix is found according to Algorithm 2, with the superscript ^{u} denoting a point on the unstable manifold. Subsequently, x_0 is both incremented and decremented by x_{step} and y_+^g/y_-^g are approximated at 45°/225° angle from p_0^u. Algorithm 2 is again invoked for these guesses, and additional points within d_{tol} of the separatrix are added to the solution vector. This procedure is repeated until the full voltage swing is covered, using linear approximation from the previous points on the separatrix to provide initial guesses at the next increments of x_i.

The subroutine, presented here as Algorithm 2, receives an initial guess, $p^g = (x, y^g)$, and returns a point $p^u = (x, y^u)$ that is within d_{tol} of the separatrix. This is done by running a series of iterative transient simulations with initial conditions

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 2 Iterative Search for Separatrix

Require: Initial guess $p^g = (x, y^g)$, and tolerance (d_{tol})

1: Set initial tolerance ΔY and initial conditions: $V_0(Q) = x; V_0(QB) = y^g$
2: Run transient simulation and find convergence state.
3: Update y^g by moving it ΔY *away* from convergence state.
4: **repeat**
5: Run transient simulation and find new convergence state
6: **if** New and previous convergence states are the same **then**
7: Continue increasing y^g *away* from convergence state.
8: **else if** *new* and *previous* convergence states are different **then**
9: The separatrix is between the last two guesses, with a maximum distance of ΔY from the last guess.
10: Update y^g *toward* the previous guess $\Delta Y = \Delta Y/2$
11: **end if**
12: **until** $\Delta Y \leq d_{tol}$
13: Return the point on the separatrix: $p^u = (x, y^g)$

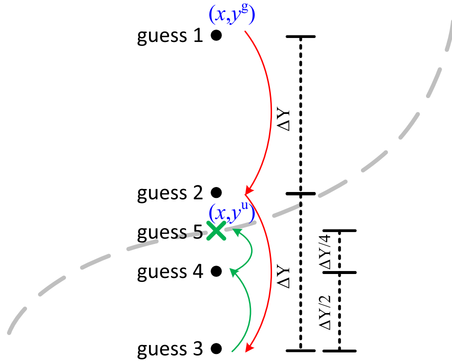


Fig. 4. Illustrative example of the operation of Algorithm 2.

that are modified according to the convergence of the previous runs. For example, if the initial guess converges to the 0 state (*above* the separatrix), the next guess is set with $y^n = y^g - \Delta Y$, where ΔY is a configurable voltage step (e.g., $0.1 \cdot V_{DD}$). An additional simulation is run with this new point. If the system converges to the opposite state (1), the separatrix is on the interval connecting the two guesses. If, however, the system converges to the same state (0), the separatrix is still below the guess, and therefore, another guess is taken with a further decreased y value. Once it is known that the separatrix is on a finite interval between two y values, a simple binary search is applied until the voltage step (ΔY) between

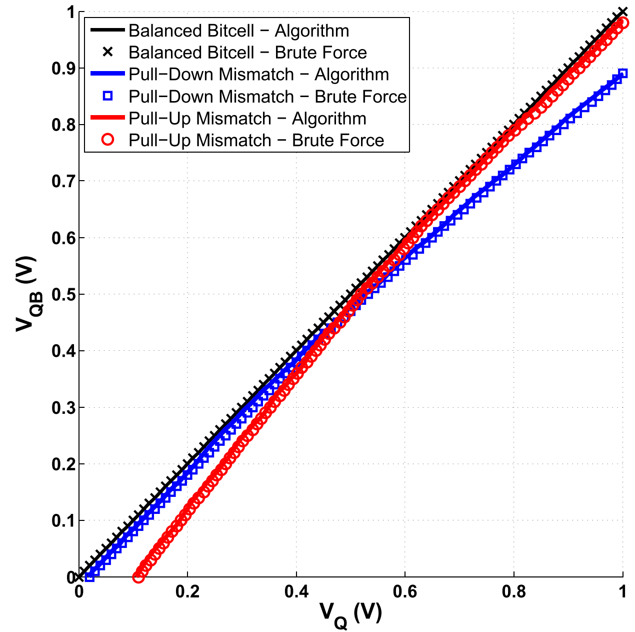


Fig. 5. Comparison of separatrix extraction with proposed algorithm to brute-force method with the same accuracy.

subsequent guesses is smaller than d_{tol}. This procedure is shown in Fig. 4.

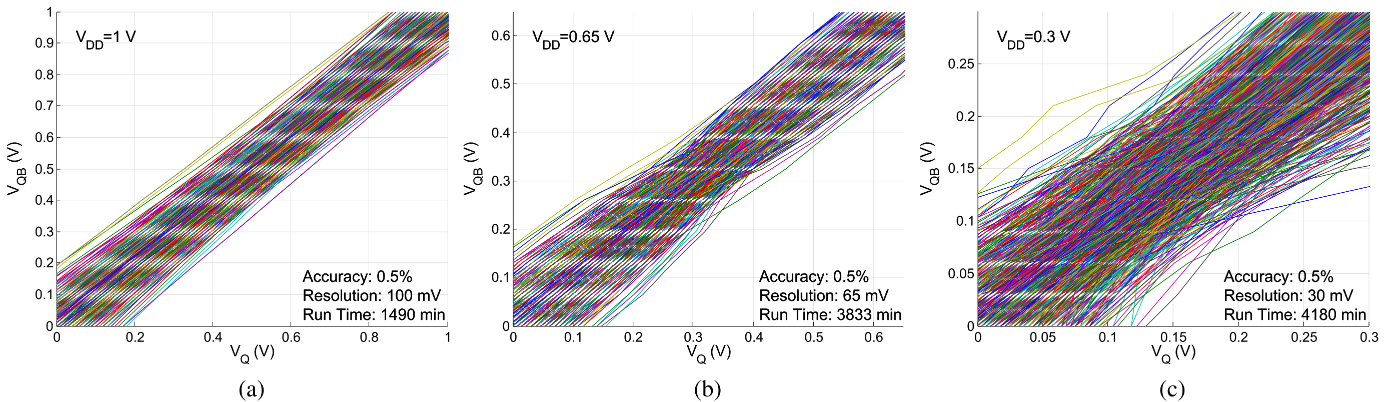
It is important to note that this algorithm is both highly configurable and enables parallelism for distributed calculation. Following an initial run on a given topology, parameters, such as y_0^g, the initial value of ΔY, the angle for the guess of the first step x_{step}, and d_{tol} can be updated to provide a sufficient result with even lower runtime. The increasing and decreasing searches (x_+ and x_-, respectively) are independent of each other, and the algorithm can be further modified by providing n initial guesses and advancing outward in $2n$ independent processes. Therefore, the overall procedure can be linearly accelerated with the use of multicored architectures.

V. Implementation Results and Statistical Distributions

The proposed algorithm was implemented as a MATLAB script running Cadence Spectre simulations on a commercial 40-nm CMOS technology based on BSIM4 models. The sampling density of our algorithm is based on two parameters: *resolution* and *accuracy*. The resolution is the distance between samples on one of the axes (e.g., distance between points on the Q axis), and the accuracy is the maximum proximity to the separatrix in direction of the other axis (QB) for each sample. For example, when searching for a point on the separatrix with $V_Q = 400$ mV with an accuracy of 1% and a 1 V supply, the resulting point [e.g., $(V_Q, V_{\text{QB}}) = (0.400, 0.523)$] is less than $1\% \cdot V_{\text{DD}} = 10$ mV away from the actual separatrix in the vertical (QB) direction. Therefore, for comparison with a 10 mV brute-force sampling grid at $V_{\text{DD}} = 1$ V, our algorithm is run with a 10-mV resolution and 1% accuracy.

The results of this comparison are shown in Fig. 5 for three standard 6T cell examples. As expected, the trivial, albeit nonphysical, example of a perfectly balanced 6T cell, displays

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) (b) (c)

Fig. 6. MC statistical distributions of the separatrices of a standard 40 nm 6T bitcell with 0.5% accuracy. Plots include 1k samples taking into consideration both global variations and local mismatch at room temperature (25 °C). Measured runtimes are for serial (single-thread) execution. (a) $V_{\mathrm{DD}} = 1$ V. (b) $V_{\mathrm{DD}} = 650$ mV. (c) $V_{\mathrm{DD}} = 300$ mV.

TABLE II
COMPARISON WITH BRUTE-FORCE EXTRACTION

Method	# Simulations	Run Time
Brute Force	10,201	755 min
Balanced Bitcell	101	7.48 min
Pull-Down Mismatch	413	30.65 min.
Pull-Up Mismatch	422	31.29 min.

a 45° separatrix for both methods. In fact, when only taking global variations into consideration, the resulting separatrix for any symmetric bitcell is very trivial. The more interesting cases are the actual, realistic circuits with some degree of mismatch between devices. For an initial demonstration of the resulting separatrix in such a case, two asymmetric examples are shown—one with 10% size mismatch between the pull-down n-type MOS devices (M1 and M4) and the other for 10% mismatch between the pull-up p-type MOS devices (M3 and M6). In these cases, the probability to find exact points on the separatrix is very small, such that the brute-force reference separatrix points in Fig. 5 represent the border between initial conditions that converged to zero and those that converged to one, or in other words, up to 10 mV from the actual separatrix. The results from our algorithm are shown to very closely match the brute-force extraction, despite requiring just 4% of the run time, as summarized in Table II.[1]

The examples in Fig. 5 emphasize that the true importance of separatrix analysis emerges under mismatch variations that cause a skew from the trivial 45° manifold. The efficiency of the proposed algorithm provides the ability to extract and display separatrices under statistical mismatch distributions, thereby providing both insight into circuit behavior and a basis for advanced dynamic stability analysis. Fig. 6(a) displays the separatrices of 1k MC samples of a 40-nm 6T SRAM bitcell with a nominal 1 V supply. This plot shows the significant skew that occurs to the RoCs of the stable states under mismatch variation. This skew increases as the supply voltage is scaled, as can be seen in Fig. 6(b) and (c) for a 650-mV and a 300-mV supply, respectively. These plots were extracted with 1% accuracy and a resolution of $0.1 \cdot V_{\mathrm{DD}}$ (i.e., 10 points per separatrix), requiring 1–4 min (using a single thread) to extract the separatrix for each sample. For comparison, using the brute-force method would require over 165 min/sample or approximately 115 d to extract Fig. 6(a) at a similar resolution and accuracy, for a speedup of over 1000×.[2]

VI. SEPARATRIX OF ALTERNATIVE SRAM TOPOLOGIES

The previous section presented our proposed algorithm and displayed the effects of device mismatch on the characteristics of the separatrix for a standard symmetric 6T SRAM bitcell. However, as extensively described in [32], this traditional circuit is limited to above-threshold supply voltages of around 0.7 V or higher. Accordingly, recent years have shown many proposals for novel SRAM bitcell circuits, most of which are based on the cross-coupled 6T circuit core, but several of which are significantly different. Separatrix extraction is a central and essential component in the analysis of such circuits; however, in the majority of these works, these curves were not presented, and none of these publications included separatrix extraction under local mismatch.

In this section, we revisit three of these alternative SRAM topologies, implementing them in a scaled 40-nm process, and plotting an MC distribution of their separatrices at subthreshold operating voltages. The goal of this discussion is to display the modularity of the proposed algorithm, demonstrating its seamless application to nonstandard topologies, while emphasizing the importance of separatrix distribution at scaled nodes and low supply voltages. We argue that all future analysis of dynamic stability should take these distributions into consideration, as they extremely influence the integrity of the results.

Schematics of the three chosen topologies are shown in Fig. 7. These topologies include the following.

[1]All run times are provided for single-threaded serial computations for an impartial comparison. Both the brute force and the proposed algorithm benefit from a linear speedup through parallelization.

[2]Note that this speedup is resolution and accuracy dependent. Whereas the brute-force method requires an increased number of simulations that is quadratically proportional to increased accuracy, the number of simulations only logarithmically increases for the proposed algorithm, and can be even lower through dynamically updating the search parameters.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

(a) (b) (c)

Fig. 7. Schematics of previously published alternative subthreshold SRAM bitcells. (a) 10T Schmitt trigger bitcell [30]. (b) Single ended 6T bitcell [31]. (c) 9T SFSRAM [18].

(a) (b) (c)

Fig. 8. MC statistical distributions of the separatrices of alternative subthreshold SRAM bitcells, implemented in a standard 40-nm CMOS process. All plots show 1000 MC samples with 0.5% accuracy and a resolution of $V_{\mathrm{DD}}/10$. (a) 10T Schmitt trigger bitcell [30]. (b) Single ended 6T bitcell [31]. (c) 9T SFSRAM [18].

1) A Schmitt-trigger-based 10T bitcell [Fig. 7(a)] proposed in [30], allowing bit interleaving and differential read, was shown to operate robustly at 160 mV in 0.13 μm CMOS.
2) A variation tolerant, single-ended 6T bitcell [Fig. 7(b)] proposed in [31], includes a complementary transmission gate and gated supplies to achieve full functionality at 193 mV in 0.13 μm CMOS.
3) A 9T supply-feedback (SFSRAM) bitcell [Fig. 7(c)] [18], utilizes internal feedback to gate V_{DD} for increased write margins and decreased leakage, and was shown to operate robustly at 250 mV in 40-nm CMOS.

These topologies were chosen, as opposed to other popular subthreshold implementations (such as [32]), as their core device composition differs significantly from the standard 6T topology, already discussed extensively in the previous section. The separatrix spreads of these bitcells, implemented in 40-nm CMOS, are shown in Fig. 8 for 1k MC samples at 0.5% accuracy. The Schmitt trigger [Fig. 8(a)] and single-ended [Fig. 8(b)] bitcells were operated at 200 mV, whereas the SFSRAM [Fig. 8(c)] was operated at 250 mV, as this was its previously presented lower functionality limit. All three plots show clearly separated RoCs, demonstrating bistability, albeit with significantly wide distributions, showing susceptibility to SEUs, as well as significantly asymmetric dynamic access behavior.

VII. Conclusion

This paper proposed a novel algorithm for SRAM separatrix tracing, a mandatory component of dynamic stability analysis. The algorithm was shown to be extremely modular, allowing simple application to any bitcell topology, while enabling simulation with accurate, industry standard technology models. For the first time, MC statistical distributions of SRAM separatrices were displayed, emphasizing the importance of these plots, especially at scaled technologies and under low supply voltages. The proposed algorithm provides as much as a 1000× speedup, as compared with traditional brute-force separatrix tracing, with many configuration options and parallel computing capabilities.

Acknowledgment

The authors would like to thank O. Sharon for his intensive work in developing the algorithm. They would also like to thank Prof. A. Fish and Prof. A. Burg for their advice and mentoring in carrying out this paper.

References

[1] (2013). *International Technology Roadmap for Semiconductors*. [Online]. Available: http://www.itrs.net
[2] E. I. Vatajelu, G. Panagopoulos, K. Roy, and J. Figueras, "Parametric failure analysis of embedded SRAMs using fast & accurate dynamic analysis," in *Proc. 15th IEEE Eur. Test Symp. (ETS)*, May 2010, pp. 69–74.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[3] B. Zhang, A. Arapostathis, S. Nassif, and M. Orshansky, "Analytical modeling of SRAM dynamic stability," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2006, pp. 315–322.

[4] L. Ding and P. Mazumder, "Dynamic noise margin: Definitions and model," in *Proc. 17th Int. Conf. VLSI Design*, 2004, pp. 1001–1006.

[5] W. Dong, P. Li, and G. M. Huang, "SRAM dynamic stability: Theory, variability and analysis," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2008, pp. 378–385.

[6] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, vol. 22, no. 5, pp. 748–754, Oct. 1987.

[7] J. Lohstroh, "Static and dynamic noise margins of logic circuits," *IEEE J. Solid-State Circuits*, vol. 14, no. 3, pp. 591–598, Jun. 1979.

[8] J. S. Yuan and L. Yang, "Teaching digital noise and noise margin issues in engineering education," *IEEE Trans. Educ.*, vol. 48, no. 1, pp. 162–168, Feb. 2005.

[9] G. M. Huang, W. Dong, Y. Ho, and P. Li, "Tracing SRAM separatrix for dynamic noise margin analysis under device mismatch," in *Proc. IEEE Int. Behavioral Modeling Simulation Workshop (BMAS)*, 2007, pp. 6–10.

[10] M. Sharifkhani and M. Sachdev, "SRAM cell stability: A dynamic perspective," *IEEE J. Solid-State Circuits*, vol. 44, no. 2, pp. 609–619, Feb. 2009.

[11] M. Wieckowski *et al.*, "A black box method for stability analysis of arbitrary SRAM cell structures," in *Proc. Conf. Design, Autom. Test Eur.*, 2010, pp. 795–800.

[12] J. Wang, S. Nalam, and B. H. Calhoun, "Analyzing static and dynamic write margin for nanometer SRAMs," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2008, pp. 129–134.

[13] Y. Zhang, P. Li, and G. M. Huang, "Separatrices in high-dimensional state space: System-theoretical tangent computation and application to SRAM dynamic stability analysis," in *Proc. 47th ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2010, pp. 567–572.

[14] R. Garg, P. Li, and S. P. Khatri, "Modeling dynamic stability of SRAMS in the presence of single event upsets (SEUs)," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2008, pp. 1788–1791.

[15] E. I. Vatajelu, Á. Gómez-Pau, M. Renovell, and J. Figueras, "Transient noise failures in SRAM cells: Dynamic noise margin metric," in *Proc. 20th Asian Test Symp. (ATS)*, Nov. 2011, pp. 413–418.

[16] J. Mezhibovsky, A. Teman, and A. Fish, "State space modeling for sub-threshold SRAM stability analysis," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2012, pp. 1823–1826.

[17] A. Teman, A. Mordakhay, and A. Fish, "Functionality and stability analysis of a 400 mV quasi-static RAM (QSRAM) bitcell," *Microelectron. J.*, vol. 44, no. 3, pp. 236–247, 2013.

[18] A. Teman, L. Pergament, O. Cohen, and A. Fish, "A 250 mV 8 kb 40 nm ultra-low power 9T supply feedback SRAM (SF-SRAM)," *IEEE J. Solid-State Circuits*, vol. 46, no. 11, pp. 2713–2726, Nov. 2011.

[19] A. Teman, A. Mordakhay, J. Mezhibovsky, and A. Fish, "A 40-nm sub-threshold 5T SRAM bit cell with improved read and write stability," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 12, pp. 873–877, Dec. 2012.

[20] C. Hill, "Definitions of noise margin in logic systems," *Mullard Tech. Commun.*, vol. 89, pp. 239–245, Sep. 1967.

[21] J. Lohstroh, E. Seevinck, and J. De Groot, "Worst-case static noise margin criteria for logic circuits and their mathematical equivalence," *IEEE J. Solid-State Circuits*, vol. 18, no. 6, pp. 803–807, Dec. 1983.

[22] J. R. Hauser, "Noise margin criteria for digital logic circuits," *IEEE Trans. Educ.*, vol. 36, no. 4, pp. 363–368, Nov. 1993.

[23] E. Grossar, M. Stucchi, K. Maex, and W. Dehaene, "Read stability and write-ability analysis of SRAM cells for nanometer technologies," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2577–2588, Nov. 2006.

[24] K. Takeda, H. Ikeda, Y. Hagihara, M. Nomura, and H. Kobatake, "Redefinition of write margin for next-generation SRAM and write-margin monitoring circuit," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, Feb. 2006, pp. 2602–2611.

[25] C. Wann *et al.*, "SRAM cell design for stability methodology," in *Proc. IEEE VLSI-TSA Int. Symp. VLSI Technol. (VLSI-TSA-Tech)*, Apr. 2005, pp. 21–22.

[26] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1859–1880, Dec. 2005.

[27] D. E. Khalil, M. Khellah, N.-S. Kim, Y. Ismail, T. Karnik, and V. K. De, "Accurate estimation of SRAM dynamic stability," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 12, pp. 1639–1647, Dec. 2008.

[28] J. M. Zurada, Y. S. Joo, and S. V. Bell, "Dynamic noise margins of MOS logic gates," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1989, pp. 1153–1156.

[29] H. K. Khalil, *Nonlinear Systems*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[30] 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.

[31] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "A variation-tolerant sub-200 mV 6-T subthreshold SRAM," *IEEE J. Solid-State Circuits*, vol. 43, no. 10, pp. 2338–2348, Oct. 2008.

[32] B. H. Calhoun and A. P. Chandrakasan, "A 256-kb 65-nm sub-threshold SRAM design for ultra-low-voltage operation," *IEEE J. Solid-State Circuits*, vol. 42, no. 3, pp. 680–688, Mar. 2007.

Adam Teman (S'10) received the B.Sc. degree in electrical engineering and the M.Sc. degree from the Ben-Gurion University of the Negev (BGU), Beersheba, Israel, in 2006 and 2011, respectively, and the Ph.D. degree from the Low Power Circuits and Systems Laboratory, VLSI Systems Center, BGU, in 2014.

He was a Design Engineer with Marvell Semiconductors, Santa Clara, CA, USA, from 2006 to 2007, with an emphasis on physical implementation. Since 2014, he has been a Post-Doctoral Researcher with the Telecommunications Circuits Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, under a Swiss Government Excellence Scholarship. He has authored 36 scientific papers, presented excerpts from his research at a number of international conferences, and holds three patent applications. His current research interests include low-voltage digital design, energy-efficient SRAM, NVM, and eDRAM memory arrays, low-power CMOS image sensors, and low-power design techniques for digital and analog VLSI chips.

Dr. Teman was a recipient of the BGU's Outstanding Project Award in 2011, the Yizhak Ben-Ya'akov HaCohen Prize in 2010, the BGU Rector's Prize for Outstanding Academic Achievement in 2012, the Wolf Foundation Scholarship for excellence in 2012, and the Intel Prize for Ph.D. students in 2013. His doctoral studies were conducted under a Kreitman Foundation Fellowship. He was honored with the Department of Electrical Engineering's Teaching Excellence Recognition at BGU from 2010 to 2012. His research activities under Prof. A. Burg include energy-efficient digital system implementation, low-power memory design, approximate computing, and significance-based computing for reliability and power optimization. He is an Associate Editor of *Microelectronics Journal*.

Roman Visotsky received the B.Sc. degree in electrical engineering from the Ben-Gurion University of the Negev (BGU), Beersheba, Israel, in 2013.

He joined Dolphin Integration, Grenoble, France, in 2013, where he currently holds an engineering position as an Analog-Mixed Signal Circuit Designer.

Mr. Visotsky was a recipient of the Best Project Award from the Department of Electrical and Computer Engineering, BGU, in 2013.