Tracing SRAM Separatrix for Dynamic Noise Margin Analysis under Device Mismatch

Garng M. Huang, Wei Dong, Yenpo Ho, Peng Li
Department of Electrical and Computer Engineering, Texas A&M University
College Station, TX 77843
Email: {ghuang, weidong, ypho, pli}@neo.tamu.edu

***Abstract*— SRAM-based memory arrays designed in deeply scaled technologies become increasingly susceptible to soft errors. A full account of SRAM cell stability requires dynamic noise margin models that take into account the temporal behavior of noise injection mechanism. One critical component of dynamic noise margin analysis is the determination of stability boundary, or the separatrix. Different from the unrealistic assumption made in prior work, we show that the separatrix is subject to significant perturbation due to device mismatch and hence must be carefully accounted for in noise margin analysis. More importantly, by applying a rigorous nonlinear system theory, we present an efficient separatrix tracing technique that can accurately determine the separatrix via fast transistor-level transient simulation. The presented technique is shown to be up to thousands times faster than a brute-force approach.**

I. Introduction

Nanometer SRAM designs are getting more susceptible to soft errors due to the scaled supply voltage. Various noise injection mechanisms such as power supply noises, substrate noises and single event upsets (SEUs) can flip the SRAM state and produce soft errors [1], [2], [3]. As such, it is important to evaluate the stability of a given SRAM cell in design. SRAM stability can be analyzed either using static noise margins (SNM) [4] or dynamic noise margins [5]. In a static noise margin analysis, one usually finds the maximum tolerable amplitude of the voltage or current perturbation on sensitive circuit nodes. However, static analysis ignores the important temporal pattern of the injected noise and the SRAM nonlinear dynamics. For this reason, a dynamic noise margin model (DNM) has been proposed [5].

In addition to analyze the stability of a nominal SRAM design, it is also crucial to evaluate the impacts of process variations. This is because the increasing process variations in nanometer technologies can introduce significant transistor parameter fluctuations such as threshold voltage and critical dimension variations. These sources of variations can perturb the noise margin, hence must be considered to evaluate the robustness of a SRAM design. In [6], statistical process variations have been considered under the context of static noise margin. In principle, the simplified closed-form dynamic noise margin model of [5] can also take process variations into account.

While dynamic noise margin analysis provides a more realistic stability analysis than its static counterpart, it also becomes more involved. In the former, one important SRAM characteristics to be analyzed is the $stability\ boundary$. During the operation of a SRAM cell, if a stable state is perturbed across that boundary, a state flipping, hence an error, will be resulted. For a perfectly symmetric SRAM, the stability boundary can be simply defined by passing a 45 degree line through the origin on the phase portrait of the SRAM cell. The stability boundary of a SRAM is also called $separatrix$ because the stability boundary separates two stability regions [7], [8]. In an SRAM cell, if the injected noise is high enough, the state of the cell can deviate from the initial stable equilibrium and go across the separatrix. If this happens, the cell state will fall into the stability region of the other stable equilibrium state and result in a state flip.

Process variations in highly scaled processes inevitably introduce device parameter variations and mismatch, hence, bring asymmetry into SRAMs. Unlike the assumption made in [5], which assumes that the separatrix remain the same even under device mismatch, the separatrix of an asymmetric SRAM cell can significantly deviate from the ideal case. Neglecting the perturbation of the separatrix can lead to inaccuracy in dynamic noise margin analysis. However, analyzing separatrix in this general case becomes rather involved and requires a full consideration of the complex nonlinear dynamics of the SRAM. In practice, a simulation-based brute-force approach does exist. However, it is based on expensive sampling of the two-dimensional state space by performing a large number of transient simulations.

In this paper, we rigorously apply a nonlinear system theory [9] to compute the separatrix. Our system-theoretic approach leads to a highly efficient approach to trace the separatrix for an asymmetric SRAM cell with the SPICE level accuracy. Our approach allows us to compute the complete stability boundary (separatrix) by running only two transient analyses, performed on a $modified$ set of transistor-level circuit equations. The proposed approach has been implemented in a transistor-level SPICE-like simulator and can lead to up to thousands times speedup for the separatrix computation compared to the brute-force method.

II. Phase portrait of SRAM cell

A standard 6-T SRAM cell is shown in Fig. 1, where the output voltage (V_1/x_1) and its complement (V_2/x_2) form the state space vector of the nonlinear dynamical system. The behavior of the SRAM cell can be described using the

978-1-4244-1567-0/07/$25.00 © 2007 IEEE

following state equations:

$$\begin{cases} \partial x_1(t)/\partial t = f(x_1, x_2) \\ \partial x_2(t)/\partial t = g(x_1, x_2) \end{cases}, \quad (1)$$

where $f(\cdot)$ and $g(\cdot)$ are certain nonlinear functions describing circuit nonlinearities.

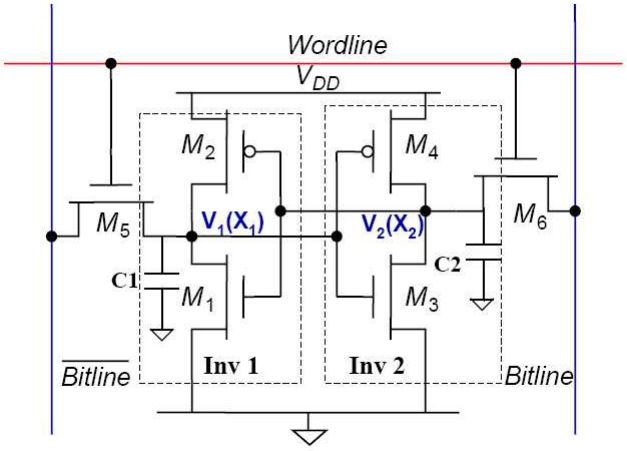

Fig. 1. A 6-T SRAM cell.

A phase portrait on the two-dimensional $x_1 - x_2$ state space is depicted in Fig. 2. Assuming a supply voltage of 1V, there are two stable equilibrium points in the phase portrait: P1($x_1 = 1,\ x_2 = 0$) and P2($x_1 = 0,\ x_2 = 1$), corresponding to the one and zero states. There also exists a meta-stable equilibrium P3, which lies around the center of the phase plot. In the standby mode, the state of the SRAM resides either at P1 or P2. If certain noise is injected, the state may be pushed away from either stable equilibrium. However, as long as the noise is small enough, the state trajectory may be still within the stability region of the initial stable equilibrium. This implies that when the noise vanishes, the SRAM state will be attracted back to the same stable equilibrium, thereby retaining the same or correct SRAM output. However, larger noises may drive the state to go across the stability boundary, and then a state flip takes place. The stability boundary, or separatrix, which separates the two stability regions, is shown in Fig. 2. Here, the SRAM cell is assumed to be perfectly symmetrical, i.e. the two cross-coupled inverters are identical. As may be expected, in this case, the separatrix is a 45 degree line passing through both the origin and P3 on the phase portrait.

However, due to unavoidable process variations, device mismatch can be introduced across the inverter pair. The separatrix of an asymmetric SRAM cell is depicted in Fig. 3 a). As can be seen, the separatrix can deviate significantly from the symmetrical case. The slope of the linear approximation (symmetrical case) does not match the true separatrix and the error would be introduced in the top right region and bottom left region. Furthermore, the separatrix in Fig. 3 a) does not appear to be a straight line and the separatrix is showing curvature as it goes through P3. In general, the separatrix can be quite nonlinear. It is important to note that such deviation

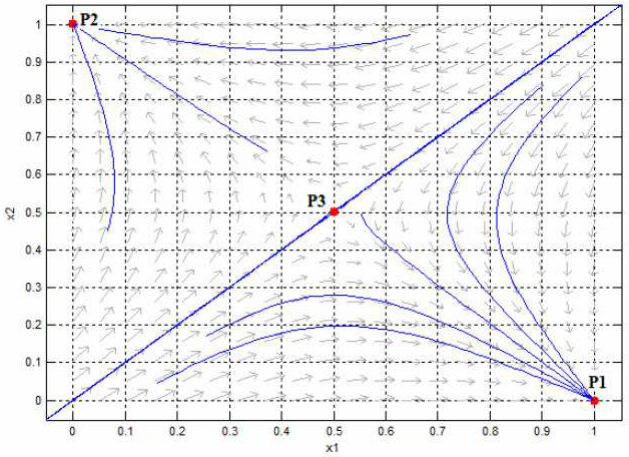

Fig. 2. The phase portrait for an SRAM cell.

from the linear approximation plays an important role in dynamic noise margin. In dynamic noise margin analysis, when appropriate, one is concerned with determining the minimum noise amplitude/duration (or energy) that leads to a state flipping. Assuming that the initial stable equilibrium is the zero state, this implies that one would like to find the minimum noise injection that can put the SRAM state right on the separatrix staring from P1, as shown in Fig. 3 a). Obviously, any shift of the separatrix will change this minimum noise level, hence, the dynamic noise margin.

In Fig. 3 b), a brute-force state space sampling method for finding the separatrix is illustrated. Essentially, the 2D state space is densely sampled using a grid, then each grid point is treated as an initial condition and a SPICE transient simulation is performed. Due to the nonlinear positive feedback loop in the cell, eventually each transient state trajectory will end up at either of the two stable equilibriums. The grid points that end up at a particular stable equilibrium correspond to the stability region of that equilibrium. The separation between the two stability regions gives the separatrix. Although a dense grid will lead to an accurate estimation of the separatrix, this method is computationally inefficient due to the large number of transient simulation runs required.

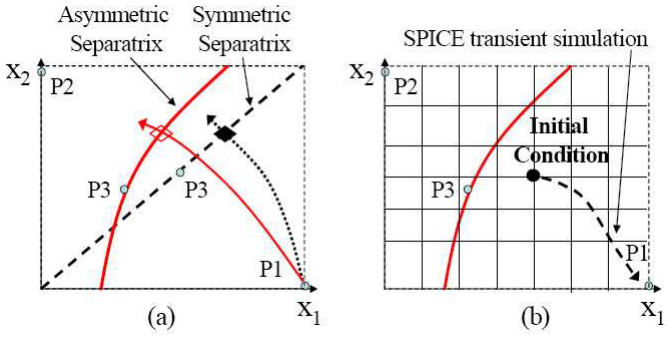

Fig. 3. a) Perturbation of the separatrix and its impact on dynamic noise margin, and b) determination of the separatrix by densely sampling the state space.

III. System-theoretical analysis

In this section, we'll more formally examine the notion of stability boundary from a nonlinear system-theoretical point of view. Finally, we'll present an efficient separatrix tracing method based on nonlinear system theory.

A. The Definition of Stability Boundary

Generally, a given dynamic equation $\dot{x} = f(x)$ with x in an N dimensional space, we can define the equilibrium points as all the $x_e's$ that satisfy $f(x_e) = 0$. Its stable manifold and stability region can be defined as below:

1) Stable Manifold: According to [9], the stable manifold is defined as:

$$W^s(x_e) = \{x \in R^N | \lim_{t\to\infty} \phi(t, x) = x_e\}, \tag{2}$$

where $\phi(t, x)$ is the trajectory that starts from x and eventually converges to x_e. As a special case, the stability region or region of attraction $A(x_e^s)$ of a stable equilibrium point x_e^s is defined to be the stable manifold of stable equilibrium point x_e^s.

In Fig. 2, P1 and P2 are two stable equilibrium points x_e^s. The stability region of equilibrium point P1 is the region of all initial states whose trajectories will converge to P1. Accordingly, the stable region of P1 is the bottom right region in the phase portrait. Likewise, the stability region of P2 is the top left region of the phase portrait. The question remains on how to describe the stability region in a precise mathematical sense. From the same figure, we can see the stability boundary (the manifold passing through P3) naturally divides the state space into two stability regions. Accordingly, the stability boundary becomes one of the key components that decide the stability margin.

2) Stability Boundary and the Separatrix: The stability boundary of the stability region is denoted by ∂A. Based on some generic conditions, which the SRAM circuit satisfies, we have the stability boundary theorem [9], in which the stability boundary can be described as:

$$\partial A = \overline{\bigcup W^s(x_e^u)}, \tag{3}$$

where x_e^u are the unstable equilibrium points on the boundary of A. Accordingly, to find the stability boundary, we need to identify the unstable equilibriums on the stability boundary and then find their stable manifolds. According to the Stable Manifold Theory [7], the stable eigenvectors of the linearized system around the equilibrium point will be tangent to its corresponding stable manifold. Thus, we can start in a small neighborhood of x_e^u along the stable eigenvector directions to integrate reverse in time to find the stable manifolds. We need to reverse in time to bypass the stability nature of the trajectories that will converge to x_e^u in a short distance.

As an example, Fig. 2 illustrates the above theorem. In Fig. 2, P3 is an unstable equilibrium point. The trajectory passes through P3 is the separatrix that separates the state space into two stability regions. Points initially starts on the Separatrix will converge to the unstable equilibrium point, P3. The tangent vector on the Separatrix is the stable eigenvector of the stable eigen-value of the linearized system around P3.

B. The Algorithm to Trace the Separatrix

Based on the stability boundary theorem and the stable manifold theorem, we can see that for a two dimensional nonlinear systems such as SRAM, the stability boundary can be found by the following procedure:

1) Find all the x_e^u and x_e^s.
2) Focus on the interested x_e^s.
3) Check if x_e^u are on the stability boundary.
4) Find the stable eigenvectors V_s of the equilibrium point x_e^u, where the stable eigenvector is the eigenvector corresponding to the stable eigen-value.
5) Choose initial conditions as $x_0 = x_e^u \pm \varepsilon V_s$, where ε is a small positive number and εV_s represents a small state perturbation.
6) Integrate backward by $\dot{x} = -f(x)$.

To find the equilibrium point, x_e, of a given nonlinear system, the following equations are solved [9]:

$$\begin{cases} \dot{x}_1 = f(x_1, x_2) = 0 \\ \dot{x}_2 = g(x_1, x_2) = 0 \end{cases}. \tag{4}$$

For a typical model, there would be three equilibrium points under zero noise condition.

To analyze the stability of a particular equilibrium point, the Jacobian matrix is evaluated at that point, and the eigen-values of the Jacobian matrix reveal the local stability around the equilibrium point. The Jacobian matrix is defined to be the following [9]:

$$J = \begin{bmatrix} \frac{\partial}{\partial x_1} f & \frac{\partial}{\partial x_2} f \\ \frac{\partial}{\partial x_1} g & \frac{\partial}{\partial x_2} g \end{bmatrix}_{x_e} \tag{5}$$

For SRAM, there would be three Jacobian matrices for three equilibrium points. The eigen-values of the Jacobian matrix evaluated at a stable equilibrium point would be both negative. The eigen-values of the Jacobian matrix evaluated at an unstable equilibrium point would be having one positive eigen-value and one negative eigen-value. Focus on the Jacobian matrix evaluated at the unstable equilibrium point. The eigenvector corresponds to the negative eigen-value is considered to be the stable eigenvector of the unstable equilibrium point. This stable eigenvector is the V_s described in step 4.

Since any point on the separatrix will converge to the unstable equilibrium point, by integrating backward from the unstable equilibrium point as described in step 6, the separatrix can be traced out from the initial starting point. In practice, we can bypass procedures 4 and 5 as long as the initial conditions are close to the unstable equilibriums since the unstable components will dissipate fast as we integrate reverse in time. Also step 3 can be skipped for SRAM cases since the unstable equilibrium point is always on the stability boundary.

IV. Implementation Issues

The proposed separatrix tracing algorithm is implemented as a part of transistor-level SPICE-like circuit simulator using C++ running on Linux operating system. As described in Section 3, to find the separatrix, the unstable equilibrium point of a SRAM design must be computed first. This is accomplished by performing a nonlinear DC circuit analysis. Care must be taken to select a proper initial guess to guarantee that the unstable equilibrium can be found by running the DC analysis.

Once the unstable equilibrium (P3 in Fig. 2) is found, a small perturbation Δp is introduced around the unstable equilibrium and the resulting state is selected as the initial transistor-level transient analysis. As described in our algorithm, the transient analysis is not applied to the original SRAM cell, instead, a modified system, which corresponds to the state equation: $x' = -f(x)$. In our transistor-level circuit simulator implementation, this corresponds as adding a minus sign to the circuit element stampings of all energy storage elements (i.e. capacitors). Therefore, the implementation of the proposed algorithm only requires some minor modification of the standard SPICE-like simulator code. The aforementioned transient simulation traces though the half of the separatrix. To find the other half, a small perturbation in the opposite direction $(-\Delta p)$ is introduced around unstable equilibrium. And a second transient analysis is conducted. As can be seen, our new algorithm only requires one standard DC analysis followed by two transient analyses, hence is much more efficient than the brute-force method.

V. Experimental Results

In our experiments, we use a level-3 SPICE device model for circuit simulation. The device model parameters are listed in Table. I. The supply voltage is set to be 1V.

TABLE I

45nm process parameters

Type	U_0	T_{ox}	Gamma	Theta
NMOS	0.05255	1.75e-9	0.2	0.5
PMOS	0.00696	1.85e-9	0.2	0.3

In order to verify the accuracy and efficiency of our proposed algorithm for tracing separatrix, we first use the algorithm to find the separatrix of a nominal (symmetric) SRAM design as shown in Fig. 1. The meta-stable equilibrium point (P3) is determined to be (0.4465, 0.4465) from the DC analysis. Then by adding a perturbation (Δp) of $\pm 1\%$, the separatrix is traced using the new algorithm. The runtime is about 1 minute in total. And the result is well-matched with what is predicted by the brute-force method. Since our separatrix tracing algorithm is conducted on the basis of modified transient analysis, the accuracy of our proposed approach is mainly determined by the time step control in a typical SPICE-like simulator. On the contrary, in addition to the time step control, the accuracy of the brute-force method is also confined by the number of grids used to sample the state space. The latter one is a much more dominant factor. Assuming that a 100x100 grid is used, and then we need to run 10,000 transient simulations to obtain the separatrix at 1% accuracy (10mv). In our experiments, one transient simulation costs about 14 seconds. Then, the total cost of 10,000 transient simulations for brute-force method is about 38 hours. Obviously, compared with the brute-force method, our proposed method is much more efficient.

After analyzing the stability of the nominal case, we now evaluate the impacts of process variations. First, we only consider the threshold voltage variations. In Fig. 4, the threshold voltages of N-type transistors M1 and M3 are simultaneously increased by 30% and the threshold voltages of P-type transistors M2 and M4 are decreased by 30%. In Fig. 5, the threshold voltages of transistors M1 and M3 are simultaneously decreased by 30% and the threshold voltages of transistors M2 and M4 are increased by 30%.

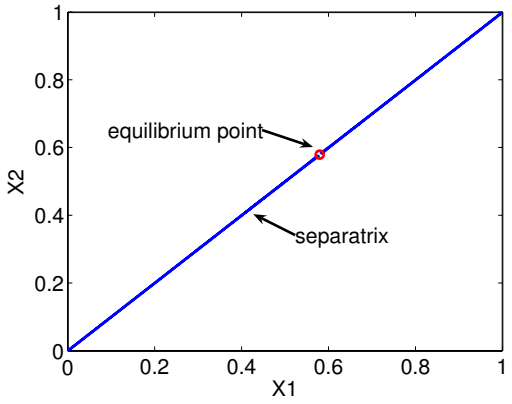

Fig. 4. Case 1: separatrix under V_T variations.

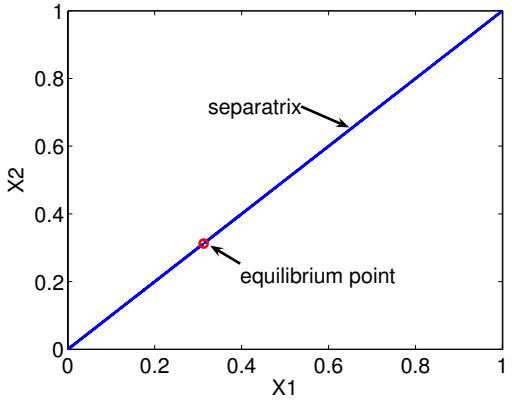

Fig. 5. Case 2: separatrix under V_T variations.

For both cases, we can observe that, the specific V_T variations introduced do not alter, the separatrix, i.e. it is still a 45 degree line passing through the origin. However, the unstable equilibrium points are shifted to (0.5801, 0.5801) and (0.3129, 0.3129) respectively for the two cases.

Next, in Fig. 6, we decrease the threshold voltage of N-type transistor M1 by 30% and increase the threshold voltage of N-type transistor M3 by 30%. At the same time, we decrease the threshold voltage of P-type transistor M2 by 30% and increase the threshold voltage of N-type transistor M4 by 30%. We employ the same tracing algorithm for separatrix in this experiment. The meta-stable equilibrium point is found to be (0.4668, 0.4141). Also we have observed the significant nonlinearity of the separatrix and the obvious deviation from the ideal symmetric case.

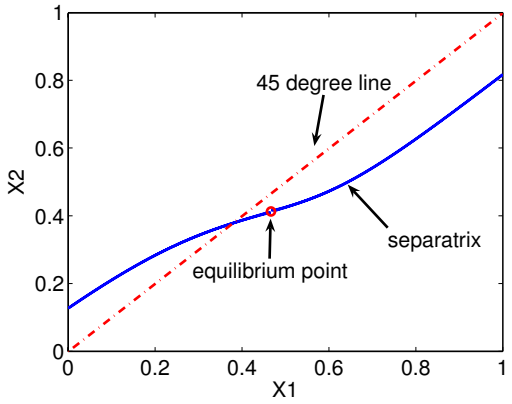

Fig. 6. Case 3: separatrix under V_T variations.

Finally, we consider transistor threshold and effective channel length variations simultaneously. We increase the threshold voltage of N-type transistor M1 by 30% and decrease the threshold voltage of N-type transistor M3 by 30%. At the same time, we increase the threshold voltage of P-type transistor M2 by 30% and decrease the threshold voltage of N-type transistor M4 by 30%. Besides these variations, the effective transistor lengths of M1, M2 are increased by 30% and those of M3, M4 are decreased by 30%. The meta-stable equilibrium point is found to be (0.3944, 0.4989). And the separatrix is shown in Fig. 7.

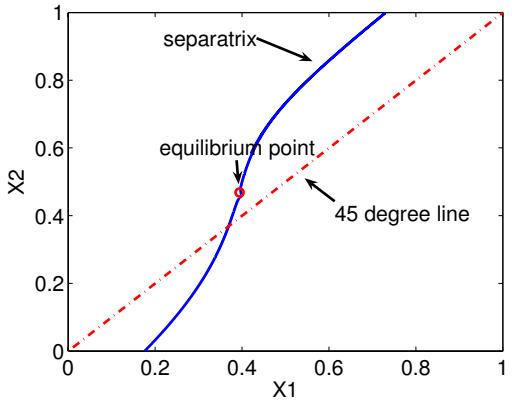

Fig. 7. Case 4: separatrix under both V_T and L_{eff} variations.

To verify the accuracy of the computed separatrix, we consider the SRAM example in Fig. 6. In order to avoid the time consuming grid-based brute-force method, we choose the following more efficient method to verify our result. We select 20 points close to the separatrix computed by the proposed algorithm, with 10 points on each side, as shown in Fig. 8. Then we run transient simulations by taking these points as initial values. We have observed that each transient state trajectory ends up at the correct stable equilibrium without crossing the separatrix, which confirms the good accuracy of the computed separatrix.

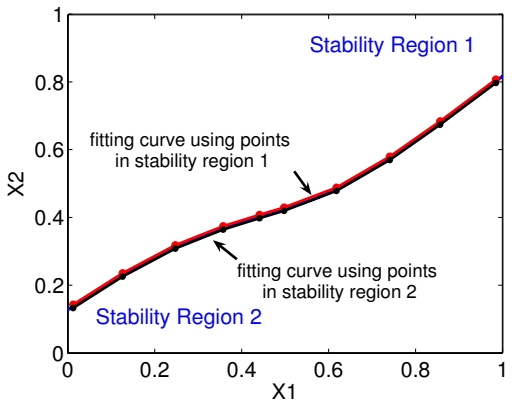

Fig. 8. Verification of the separatrix of Fig. 6.

VI. Conclusion

In this paper, an efficient SRAM stability boundary (separatrix) tracing algorithm is presented. Based on a rigorous nonlinear system theory, our algorithm can quickly find the non-ideal separatrix by performing fast transient analysis based tracing. Our technique provides useful insight to SRAM dynamic stability margin when process variation introduced asymmetry is present.

References

[1] H. H. Chen and J. S. Neely. Interconnect and circuit modeling techniques for full-chip power supply noise analysis. *IEEE Trans. Components, Packaging, and Manufacturing Technology-Part B*, pages 209–215, 1998.

[2] R. Gharpurey and R. G. Meyer. Modeling and analysis of substrate coupling in ic's. In *IEEE Custom-Integrated Circuit Conf.*, pages 125–128, 1995.

[3] P. E. Dodd and L. W. Massengill. Basic mechanisms and modeling of single-even upset in digital microelectronics. *IEEE Trans. Nucl. Sci.*, pages 583–602, 2003.

[4] E. Seevink, F. List, and J. Hohstroh. Static-noise margin analysis of mos sram cells. *JSSC*, pages 784–754, 1987.

[5] B. Zhang, A. Arapostathis, S. Nassif, and M. Orshansky. Analytical modeling of sram dynamic stability. In *Proc. of IEEE/ACM ICCAD*, Nov 2006.

[6] K. Agarwal and S. R. Nassif. Statistical analysis of sram cell stability. In *Proc. of IEEE/ACM DAC*, pages 57–62, 2006.

[7] H. Khalil. *Nonlinear Systems*. Prentice-Hall, 2002.

[8] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and bifurcation of Vector Fields*. Springer Verlag, 1983.

[9] J. Zaborszky, G. Huang, B. Zheng, and T.C. Leung. On the phase portrait of a class of large nonlinear dynamic systems such as the power system. *IEEE Transactions on Automatic Control*, 1988.