Loop flattening & spherical sampling: Highly efficient model reduction techniques for SRAM yield analysis

Loop Flattening & Spherical Sampling: Highly Efficient Model Reduction Techniques for SRAM Yield Analysis

Masood Qazi, Mehul Tikekar, Lara Dolecek, Devavrat Shah, and Anantha Chandrakasan
Massachusetts Institute of Technology, Cambridge, MA
Email: mqazi@mit.edu, mtikekar@iitb.ac.in, dolecek@ee.ucla.edu, devavrat@mit.edu, anantha@mtl.mit.edu

Abstract— **The impact of process variation in deep-submicron technologies is especially pronounced for SRAM architectures which must meet demands for higher density and higher performance at increased levels of integration. Due to the complex structure of SRAM, estimating the effect of process variation accurately has become very challenging. In this paper, we address this challenge in the context of estimating SRAM timing variation. Specifically, we introduce a method called *loop flattening* that demonstrates how the evaluation of the timing statistics in the complex, highly structured circuit can be reduced to that of a single chain of component circuits. To then very quickly evaluate the timing delay of a single chain, we employ a statistical method based on importance sampling augmented with targeted, high-dimensional, *spherical sampling*. Overall, our methodology provides an accurate estimation with 650X or greater speed-up over the nominal Monte Carlo approach.**

I. Introduction

The performance evaluation of circuit designs, SRAM in particular, is becoming an increasingly difficult task due to the process variation in deep-submicron technologies. The complex structure of SRAM—which is an assembly of multiple components (memory cells, sense amplifiers, delay chains) at different rates of repetition—makes it very challenging to estimate the effect of process variation on critical circuit properties, such as the timing delay.

In such scenarios, the analytical distribution of the performance metrics is not known. As a consequence, any statistical simulation method unavoidably resorts to numerical solvers such as SPICE. Classical approaches like the Monte Carlo methods require too many iterations of such SPICE evaluations because of the circuit complexity and extremely low tolerable failure probabilities of individual components (10^{-8} and below). Thus, the primary challenges to any statistical simulation method are: (a) dealing with the structural complexity of the timing delay evaluation problem, and (b) estimating timing delay statistics to a very high accuracy. In this paper, we shall overcome these two challenges for the timing delay analysis of SRAM by means of two proposed methods of *Loop flattening* and *Spherical Importance Sampling*, respectively.

Prior work. A lot of exciting recent work has made important progress towards the eventual goal of designing generically applicable, efficient simulation methodologies for circuit performance evaluation. However, this line of work uniformly falls short in addressing the above stated challenges regarding the evaluation of timing delays of integrated SRAMs.

To begin with, in [1] and [2] authors have developed efficient sampling based approaches that provide significant speedup over the Monte Carlo. However these works do not deal with the interconnection complexity, i.e., do not address the challenge (a) stated above. And hence, as is, they suffer from the curse of dimensionality that results from a very large number of transistors in relevant circuit applications.

As noted earlier, the main bottleneck for efficient simulation is the utilization of a circuit solver like SPICE, as this approach is necessitated by the lack of an exact analytic description of the variation of performance metrics. In [3], by modeling the bitline signal and the sense amplifier offset (and the timer circuit) with Gaussian distributions, the authors are able to come up with a linearized model for the read path. As this model can be simulated in MATLAB, the SRAM structure can be emulated and the evaluation time can be significantly improved. This approach, though interesting, is very limited since an exact analytic description is unlikely to be known in general, or even considered as a truthful approximation of reality. It is then only reasonable to look for a generic simulation method that can work with *any* form of distributions implied by the circuit solver such as SPICE.

Finally, in [4] the authors have extended upon the conventional worst-case approach to SRAM design in which the largest offset sense amplifier is required to support the weakest memory cell. Their proposed method requires that an underlying Gaussian distribution models the bitcell current—particularly in the extreme tails of the distribution—in order to enable the numerical convolution of a Gumbel distribution of the worst-case read current with the sense amplifier offset. Nevertheless their work does not put forth an unambiguous approach as it requires the designer to heuristically allocate margin between memory cells and sense amplifiers.

Our contributions. As the main result of this paper, we put-forth a statistical methodology to quickly and accurately evaluate timing delay in an SRAM. The two key aspects of our methodology are: (i) the *Loop flattening* method and (ii) the *Spherical Importance Sampling*.

The loop flattening approach addresses the added difficulty of the timing delay estimation caused by the complex structure of SRAM. We show that, surprisingly, the naive adaptation of the classical critical path methodology—in which a Monte Carlo simulation of a chain of component circuits (such as row driver, memory cell, and sense amplifier) disregards the relative rate of replication—produces an estimate that is *always* conservative and highly accurate. Namely, if the loop flattening based approach indicates that the delay exceeds

978-3-9810801-6-2/DATE10 © 2010 EDAA

130ps with probability 10^{-5}, then the actual delay will exceed 130ps with probability less than or equal to 10^{-5}. More importantly, unlike the worst-case estimation, this *conservative approach is increasingly accurate at lower failure levels*.

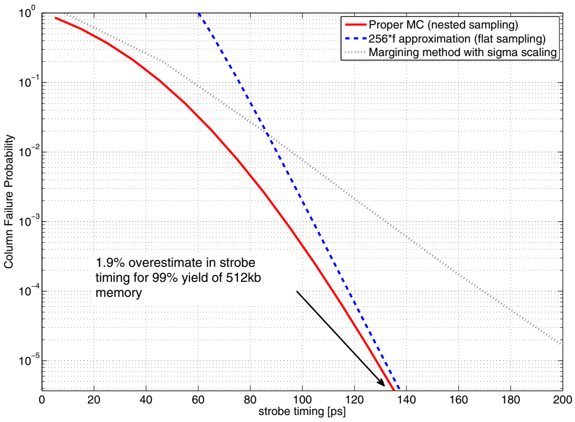

Fig. 1. The conventional Monte Carlo (nested sampling) estimation and the proposed loop flattening technique are compared in a simplified read path example in MATLAB. The delays associated with the two curves for a common level of failure tightly converge with the decreasing failure rate. High accuracy is achieved at relevant (low) levels of failure.

The reduction obtained by loop flattening still leaves us with the problem of evaluating the probability density function (pdf) of a single chain of SRAM components (a single memory cell and a single sense amplifier), to a very high accuracy. This problem requires sampling a 12-parameter distribution, wherein a standard Monte Carlo is clearly not useful. The importance sampling -based approach in the recent work [2] utilizes 'uniform sampling,' and as such does not scale to higher dimensionality (12 dimensions in the case of interest). To cope with the dimensionality, we use a spherical sampling based approach that samples space in an adaptive manner.

The combination of these two methods, loop flattening and Spherical Importance Sampling, yields an efficient and an accurate approach for estimating the timing delay in SRAM. As a representative application of our method, consider Fig. 1. Given a specific setup ($1mv/ps$ bitline discharge with standard deviation of 10%, $20mV$ of sense amplifier offset, and 256 cells per bitline), the curves of sense amplifier timing (X-axis) versus the probability of incorrect sense amplifier output (Y-axis) are shown for: 1) exhaustive Monte Carlo (solid red), 2) the worst-case estimation (dashed gray) and 3) our approach (dashed blue). We make the following observations based on this figure. First, our estimation is always conservative, as expected. Second, it becomes highly accurate for low probability of error, which is the regime of interest. For example, for the probability of error of 10^{-5} of a single memory column, the timing delay prediction of our method is off by 1.9% compared to the Monte Carlo approach. This corresponds to the 99% yield of a modest 512kb memory. Third, the worst-case estimation—in which the weakest memory cell must overcome the largest sense amplifier offset—is far too conservative for this probability of error. Even in MATLAB, the exhaustive Monte Carlo simulation took over six hours; whereas, the loop flattened curve was obtained instantaneously as it arises from the sum of two Gaussians. In the experimental results that follow, our approach exhibits 650X or greater speedup over a Monte Carlo approach, while still using the more generally applicable SPICE solver.

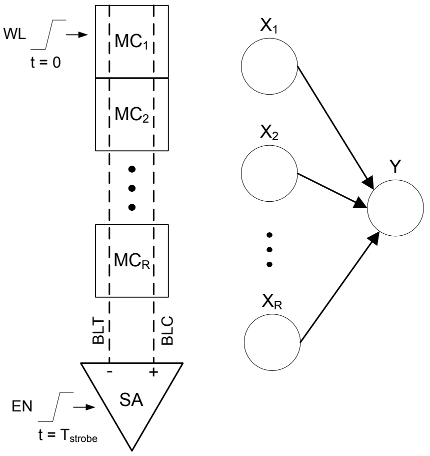

Fig. 2. A simplified critical path for small-signal sensing.

Organization. The remainder of the paper is organized as follows. Section II introduces our approach based on loop flattening and Spherical Importance Sampling. Section III provides details on its application to the SRAM. Finally, Section IV presents conclusions and directions for future work.

II. Our Approach

We describe our approach in the context of analyzing timing variation in SRAM. However, as the reader will notice, it is likely to be widely applicable due to its generality.

Setup. Fig. 2 shows a simplified critical path for small-signal sensing in a memory column. Each memory column has R memory cells and a sense amp. Fig. 3 displays a transistor level schematic of the standard memory column with $R = 128$. In an SRAM, there are M such memory columns. We are interested in the probability of incorrect output for a given strobe time setting, T. Specifically we want $F(T) = \Pr(S \leq 0)$ where S is the worst-case value of the difference between the amount of signal produced by a memory cell and the amount of offset in its supporting sense amplifier. The amount of signal produced by a memory cell scales proportionally to the allowed strobe timing, T. We describe our approach via the two main ingredients: first, the loop flattening approximation and second, Spherical Importance Sampling.

Ingredient 1. Loop Flattening. Let S_k be the worst-case signal in the kth memory column. Let $S = \min_{1 \leq k \leq M} S_k$. It is reasonable to model each S_k as independent and identically distributed. Therefore, to evaluate $F(T)$, we have

$$\begin{aligned} F(T) &= \Pr(S \leq 0) \;=\; 1 - \Pr(S > 0) \\ &= 1 - \Pr(S_1 > 0)^M \\ &= 1 - (1 - \Pr(S_1 \leq 0))^M . \end{aligned}$$

We may focus on evaluating $\Pr(S_1 \leq 0)$ for a single memory column. As shown in Fig. 2, let X_i be the rate of signal development by the ith memory cell of the column, and let Y be the sense amplifier offset. Then,

$$S_1 = \min_{1 \leq i \leq R} Z_i,$$

where $Z_i = T \cdot X_i - Y$. Therefore, using the standard union bound [5],

$$\begin{aligned} F_1(T) &= \Pr(S_1 \leq 0) \\ &= \Pr\left(\cup_{i=1}^{R}\{Z_i \leq 0\}\right) \\ &\leq R \cdot \Pr(Z_1 \leq 0) \;=\; R \cdot f_1(T), \end{aligned}$$

where $f_1(T) = \Pr(Z_1 \leq 0)$. Putting the above into the context of M independent memory columns,

$$F(T) \;\leq\; 1 - (1 - Rf_1(T))^M . \qquad (1)$$

Further, for T large enough, i.e., for $f_1(T)$ small enough (which is indeed the case of interest), $Rf_1(T)$ is small, and we obtain an approximation (using $1 - x \approx e^{-x}$ for small x)

$$F(T) \;\leq\; 1 - e^{-MRf_1(T)} \;\approx\; MRf_1(T). \qquad (2)$$

Equation (2) suggests that $F(T) \approx MRf_1(T)$. We call this the *loop flattening* approximation—it is conservative as the above derivation shows. It can be shown to be asymptotically correct, i.e.,

$$\lim_{T \to \infty} \frac{1}{T} \log \frac{F(T)}{MRf_1(T)} \;=\; 0, \qquad (3)$$

when X_i, Y have reasonable distributions such as Gaussian. Intuitively, the approximation (2) says that *a large delay primarily happens due to only one of the cells, that is, multiple cells are unlikely to simultaneously induce a large delay*. In Fig. 1 $R = 256$, X_i is a Gaussian with the mean of $1mV/ps$ and relative standard deviation of 10%, and Y_i is a zero-mean Gaussian with $20mV$ of standard deviation. The evaluation of these parameters show that the convergence of (3) is very tight at practical strobe timings.

Ingredient 2. Spherical Importance Sampling. Equation (2) suggests that we need to evaluate $f_1(T) = \Pr(Z_1 \leq 0)$ for $T \geq 0$. To do so, we propose an importance sampling based approach with spherical sampling. We quickly recall the basics of the importance sampling in the context of our setup.

Variable $Z_1 = T \cdot X_1 - Y$ is determined by the 12 parameters, say $A_\ell, 1 \leq \ell \leq 12$, that correspond to the random threshold voltage variation of transistors in the critical path (drawn in dark lines) of Fig. 3. To capture process variation in SRAM, these variables are typically modeled as independent Gaussians with means $\mu_\ell, 1 \leq \ell \leq 12$, and variances $\sigma_\ell^2, 1 \leq \ell \leq 12$. Since the relationship between A_ℓ's and Z_1 is described via an implicit function in SPICE, a simulation-based approach is a de facto method for determining $f_1(T) = \Pr(Z_1 \geq T)$. However, for large T, i.e., when $f_1(T)$ is small, the standard Monte Carlo approach would require too many samples. Fortunately, importance sampling provides a way to speed up such a simulation [6]. In a nutshell, one draws values for $A_\ell, 1 \leq \ell \leq 12$ as per Gaussian distributions with means $\mu_\ell + s_\ell, 1 \leq \ell \leq 12$, and variances, $\sigma_\ell^2, 1 \leq \ell \leq 12$, where $s_\ell, 1 \leq \ell \leq 12$ are *cleverly* chosen mean shifts so that $\Pr(Z_1 \geq T)$ becomes likely under the new shifts. Under these shifted variables, one estimates $\Pr(Z_1 \geq T)$ highly accurately, while using only a few samples, say N. (The explicit transformation between the original and the new sampling domains reverts this estimate back to the original domain at no additional cost). Precisely, for $1 \leq \ell \leq 12$, let the values of A_ℓ for these N samples be $a_\ell(n), 1 \leq n \leq N$. Let $\chi(n) = 1$ if $Z_1(n) \geq T$, and 0 otherwise. Here $Z_1(n)$ is the excess signal observed in a SPICE simulation that is fed values $a_\ell(n), 1 \leq \ell \leq 12$. Then, the importance sampling estimator of $f_1(T)$ based on the shifts $\mathrm{s} = (s_\ell)_{\ell=1}^{12}$ is given by

$$\hat{f}_1(T, \mathrm{s}) = \frac{1}{N}\left(\sum_{n=1}^{N} \chi(n)\; e^{-\sum_{\ell=1}^{12}\left(\frac{s_\ell(2a_\ell(n) - 2\mu_\ell - s_\ell)}{\sigma_\ell^2}\right)}\right). \qquad (4)$$

The non-triviality lies in finding an appropriate mean shift vector s so that the estimate $\hat{f}_1(T, \mathrm{s})$ converges quickly—that

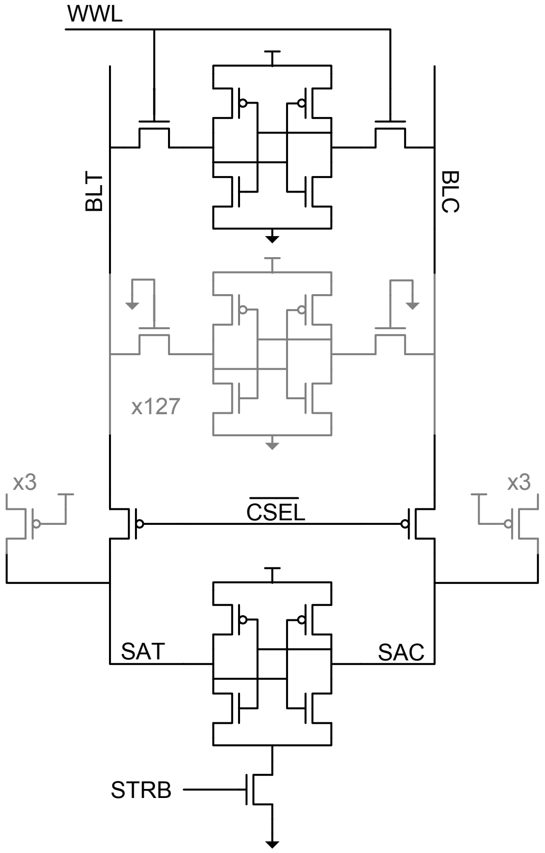

Fig. 3. Transistor level schematic of a representative memory column with 128 cells per bitline and additional multiplexing of 4 columns per sense amplifier. Precharge devices are not shown for clarity.

is, in very few samples N. The overall cost of estimating $f_1(T)$ is then the sum of (i) the number of samples required to discover a good shift s, and (ii) the number of samples required to obtain a convergent estimate $\hat{f}_1(T, \mathrm{s})$ based on the shift s.

To minimize this overall sampling cost we devise an approach, which we call Spherical Importance Sampling. Mathematically speaking, the objective is to identify a mean-shift vector s such that this vector corresponds to the point on the pass-fail boundary closest in quadratic distance to the nominal operating point, cf. Fig. 4. Such a point corresponds to the most likely failure mechanisms as recognized in Large Deviations Theory [7] and in the field of robust mechanical system engineering [8], [9].

For a uniform characterization of such an estimator, we employ a figure of merit (FOM) that ensures a predetermined level of confidence and accuracy [2].

We now outline the main steps of the overall procedure.

Step 1. Perform spherical search.

Given the initial tolerable failure floor p_{floor} from a user (e.g., 10^{-12}) initialize the algorithm parameters: R_{high} = $2\phi^{-1}(1-p_{\text{floor}})$, R_{low} = 0, and N_{iter} = 0. $\phi(\cdot)$ is the normal CDF.

For the allocated complexity parameter N_1, set L and U as the lower and upper bounds on the target number of fails. A good design choice is $U = 0.5N_1$ and $L = 1$.

(a) While $M_f \notin [L, U]$:

- Set $R_1 := 1/2(R_{\text{low}} + R_{\text{high}})$
- Sample the radius-R_1 spherical shell N_1 times, and record the total number of failures, M_f.
- If $M_f > U$, set $R_{\text{high}} := R_1$
- If $M_f < L$, set $R_{\text{low}} := R_1$
- $N_{\text{iter}} := N_{\text{iter}} + 1$.

(b) Record the final number of iterations as $B_1 = N_{\text{iter}}$. Average over the direction vectors associated with all the failing vectors in the last iteration in step (a). Initialize s_1 with this average as the starting mean-shift for Step 2. Record the quadratic norm of this shift as the current minimum norm, $minNorm$ = norm(s_1).

Step 2. Perform local exploration.

Let N_2 be the allocated complexity of this step. Set $runCount = 0$, initialize $R_2 = R_1/2$, and $\alpha = (0.05/R_2)^{\frac{1}{N_2}}$. The latter parameter governs how the resolution of local exploration will gradually try to zoom in.

While $runCount \leq N_2$:

- Sample uniformly from a spherical distribution of radius R_2, around the point s_1. Call this point s_x.
- If norm(s_x) $< minNorm$,
  - Set $runCount := runCount + 1$.
  - Run a SPICE simulation with s_x as its input. If the simulation results in a failure, record the displacement $d = \mathrm{norm}(\mathrm{s}_1 - \mathrm{s}_x)$ and then update the mean shift vector : $\mathrm{s}_1 = \mathrm{s}_x$. Otherwise $d = 0$.
  - Geometrically shrink R_2, while factoring in the displacement:

$$R_2 := \alpha R_2 + (1-\alpha)d$$

Step 3. Run the off-the-shelf importance sampler.

This step is done as per (4) with $\mathrm{s} = \mathrm{s}_1$. Record N_3 as the number of steps it takes the estimator to reach the FOM value of 0.1, corresponding to the accuracy and confidence of 90% each.

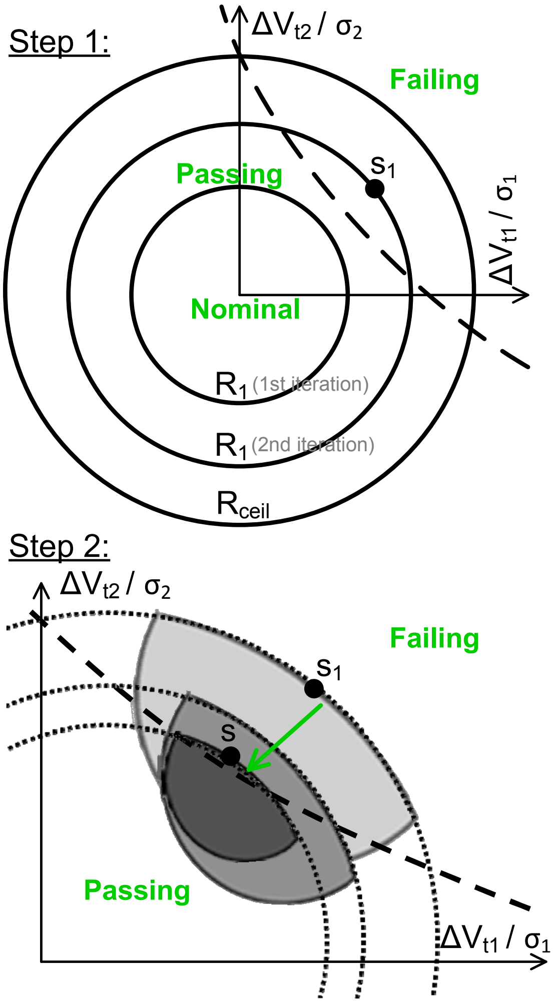

Fig. 4. Illustration of Spherical Importance Sampling Algorithm. In Step 1, samples on spherical shells quickly find a good starting direction enabling the local exploration in Step 2 to gravitate to the mean shift of minimum quadratic norm.

The key idea is that Step 1 quickly gives a coarse sense of direction of the appropriate mean-shift vector, and that Step 2 fine-tunes the vector selection across the local neighborhood in the parameter space, see Fig. 4.

The overall cost of the proposed approach is then $N_{total} = N_1 \times B_1 + N_2 + N_3$. In the next section we shall see how the proposed procedure can be effectively applied to SRAM circuit design.

III. Application to SRAM Circuit Design

The circuit in Fig. 3 is simulated in a generic 45nm technology [10] with High-k Metal Gate 45nm PTM models [11] at the nominal process corner. Based on the measurement data in [12] the local random variation of transistor threshold voltage is modeled as:

$$\sigma_{V_t} = \frac{1.8mV}{\sqrt{W_{\text{eff}} L_{\text{eff}}}}.$$

Furthermore, the memory cell has been designed and laid out to exhibit a read instability probability of less than 10^{-9}. The layout of this 6T memory cell in the freePDK 45nm technology is shown in Fig. 5; the layout gives the bitline capacitance of $0.24 fF/\mu m$ and, it also gives the physical cell dimensions of $1.54\mu m \times 0.38\mu m$. The sense amplifier and column multiplexer devices were sized to occupy the area of roughly eight memory cells.

The waveforms of a nominal Monte Carlo simulation with an aggressive timing are shown in Fig. 6. About $40mV$ of highly variable bitline signal (BLT - BLC) interacts with the sense amplifier offset to produce a logic level on the output of the sense amplifier (SAT - SAC). A larger strobe time setting, t_{STR}, allows more bitline signal to develop and therefore reduces the probability of incorrect output. The complex nature of the waveforms shows how no suitable performance metric can be defined to facilitate analytical modeling or even the formulation of numerical derivatives in the space of variation parameters. Therefore, techniques based on sensitivity analysis [13], hypersphere or directional cosine searches [9], or approximations for simplified Monte Carlo simulation [3] will significantly compromise accuracy.

On the other hand, the processing of the read path simulation results leads unambiguously to a binary indicator of pass/fail, and, in general, the formulation of a binary indicator is feasible for arbitrary circuits. In order to evaluate the probability of incorrect read-out versus strobe time setting, the proposed Spherical Importance Sampling algorithm is applied directly to this complicated simulation as it requires nothing more than a binary simulation output—1 if the sense amplifier output is incorrect and 0 otherwise.

Shown in Fig. 7 is the evolution of the failure probability estimate versus simulation run index for both the importance sampling stage (step 3) of the proposed algorithm and nominal Monte Carlo. The strobe time setting of $40ps$ results in a path failure probability of $1.01 \cdot 10^{-4}$ which, by the loop-flattening technique described in Section II, results in a column failure of $5.2 \cdot 10^{-2} = 128 \cdot 4 \cdot 1.01 \cdot 10^{-4}$. The raw speed-up of step 3 is 1800X, and taking into consideration the cost of step 1 and step 2, the total speedup is 650X. This reflects a 4.3X speedup over the work in [2] at the same level of failure probability, *and with twice the dimensionality—12 instead of 6*. The specific algorithm parameters used for all runs were: $p_{floor} = 10^{-12}$, $N_1 = 500$, $N_2 = 500$.

Shown in Table I is a summary of simulation results for four strobe time settings. One million Monte Carlo runs takes seven days on one Linux workstation utilizing 1 Intel 3.2GHz CPU with 16GB of RAM; therefore, verification with a SPICE Monte Carlo benchmark for failure probabilities of 10^{-5} or lower is infeasible as it requires ten million or more runs. For strobe timings of $50ps$, $65ps$, $80ps$, the speed-up is compared against a projected number of required Monte Carlo runs with the expression $100/p_{\text{path}}$. As additional verification, the Spherical Importance Sampling algorithm was verified at these failure levels with a linear function in MATLAB and the step 3 importance sampling run was extended 3X beyond the required number of runs to verify stable convergence of the failure estimate. Table I shows how the simulation cost gradually increases with exponentially lower levels of failure probability, achieving a speed-up of over twenty million at $p_{path} = 1.91 \cdot 10^{-9}$.

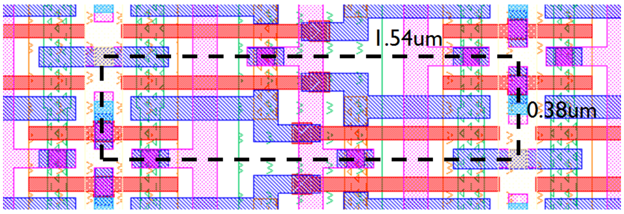

Fig. 5. The layout of the 6T memory cell in the freePDK 45nm technology. Extraction from layout reveals 0.24fF/μm of bitline capacitance.

The increasing cost comes from needing more runs in step 1 to find a suitable sampling shell radius, and from needing more runs in step 3 to achieve the desired level level of confidence. Generally, no more than two radius trials were required to get a useful direction to initialize the local exploration in step 2. As an example, Table II shows the evolution of the mean shift after completing spherical sampling ($\mathbf{s}_1$) and then after completing local exploration ($\mathbf{s}$), when evaluating the strobe timing of $80ps$. The first row shows how the spherical sampling gets a coarse direction correct along a few key dimensions (1, 3, 4, 8) and then the local exploration greatly improves the accuracy of the shift both in terms of magnitude and direction. The resulting shift, $\mathbf{s}$ in the second row matches circuit insight: 1) the column multiplexer devices have little influence; 2) devices associated with the non-discharging side (BLC) in Fig. 6 have small magnitude; 3) the read stack in the memory cell (10,12) is significantly weakened; 4) the mismatch between NMOS devices (3,4) in the sense amplifier is most critical. Going from $\mathbf{s}_1$ to $\mathbf{s}$, the mean shift for the importance sampling run, took only 500 trials with an initial exploration radius of 5.25 tapering down to 0.06. As discussed in section II, the local exploration radius can at most shrink down to a magnitude of 0.05. The actual value of the final radius being close to 0.05, indicates that most of the local exploration runs resulted in a small displacement. A larger final radius would have suggested the need for a larger complexity parameter N_2.

IV. Conclusion

In this paper we presented two techniques—*loop flattening* and *Spherical Importance Sampling*—and a method to syn-

	Sense Amplifier				Col. Mux		Memory Cell						
shift	ΔV_{t1}	ΔV_{t2}	ΔV_{t3}	ΔV_{t4}	ΔV_{t5}	ΔV_{t6}	ΔV_{t7}	ΔV_{t8}	ΔV_{t9}	ΔV_{t10}	ΔV_{t11}	ΔV_{t12}	L_2 norm
$\mathbf{s}_1$	-0.89	-1.45	-3.61	5.59	0.03	-0.93	-3.29	0.19	-3.58	6.27	-0.53	-0.28	10.55
$\mathbf{s}$	-0.65	0.35	-3.60	3.27	0.20	0.00	-0.02	0.08	-0.13	0.69	0.18	3.24	5.90

TABLE II

EVOLUTION OF MEAN SHIFT VECTOR AFTER SPHERICAL SAMPLING (STEP 1) AND LOCAL EXPLORATION (STEP 2) FOR STROBE TIMING OF $80ps$.

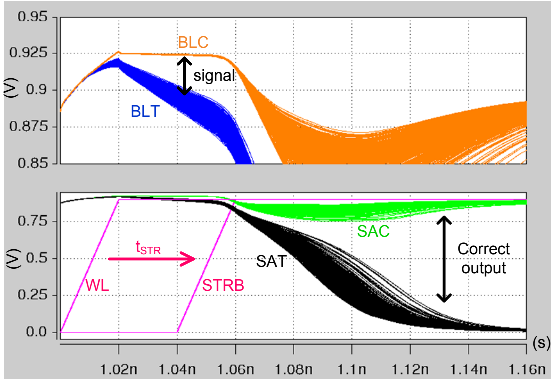

Fig. 6. Operational waveforms of the read column illustrate the pronounced effect of device variation and the difficulty in defining a performance metric conducive to analytical modeling or even the formulation of numerical derivatives.

t_{STR}	p_{path}	cost	speed-up	p_{col}
40ps	$1.01 \cdot 10^{-4}$	1534	$6.50 \cdot 10^{2}$	$5.2 \cdot 10^{-2}$
50ps	$9.08 \cdot 10^{-6}$	1660	$6.63 \cdot 10^{3}\dagger$	$4.6 \cdot 10^{-3}$
65ps	$1.33 \cdot 10^{-7}$	2214	$3.40 \cdot 10^{5}\dagger$	$6.8 \cdot 10^{-5}$
80ps	$1.91 \cdot 10^{-9}$	2423	$2.16 \cdot 10^{7}\dagger$	$9.8 \cdot 10^{-7}$

TABLE I

SUMMARY OF S-IS SIMULATION RESULTS ON SPICE SIMULATION OF THE READ COLUMN IN FIG. 3. THE PATH FAILURE PROBABILITY IS MULTIPLIED BY $512 = 128 \cdot 4$ TO PRODUCE THE OVERALL COLUMN FAILURE. SPEED-UP OVER NOMINAL MONTE CARLO IS 650X OR HIGHER. VALUES MARKED WITH †ARE PROJECTIONS ON THE NUMBER OF MONTE CARLO RUNS USING THE RULE OF THUMB $100/p_{\mathrm{path}}$.

thesize them to reduce the statistical analysis of an SRAM block to an Importance Sampling simulation of a chain of component circuits. The key challenge of searching for the most likely failure mechanism in a high dimensionality (12 in this work) parameter space is addressed by a two-stage process in which a coarse direction is obtained first, followed by a local sampling of increasing resolution.

As a contrast to prior SRAM yield analysis, the consideration of intermediate metrics (bitline signal, sense amplifier offset) has been replaced by a full-scale SPICE simulation requiring only the indication of pass or fail. As future work, this method can be extended to the complete, global row and column path of large embedded SRAM, in addition to other highly structured circuits such as adders, FIR filters, and FFT accelerators.

ACKNOWLEDGEMENT

This work was funded in part by the FCRP Focus Center for Circuit & System Solutions (C2S2), under contract 2003-CT-888.

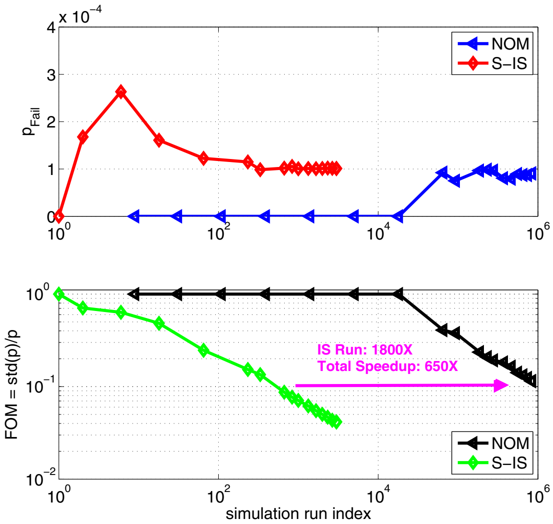

Fig. 7. Evolution of failure probability estimate from Spherical Importance Sampling (S-IS) compared with nominal Monte Carlo for a strobe time setting of $40ps$.

REFERENCES

[1] A. Singhee and R. A. Rutenbar, "Statistical blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application," in *DATE*, 2007.

[2] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: Sram evaluation through norm minimization," in *ICCAD*, 2008.

[3] M. H. Abu-Rahma *et al.*, "A methodology for statistical estimation of read access yield in srams," in *DAC*, 2008.

[4] R. Aitken and S. Idgunji, "Worst-case design and margin for embedded sram," in *DATE*, 2007.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*. Athena Scientific, 2002.

[6] J. Bucklew, *Introduction to Rare Event Simulation*. Springer, 2004.

[7] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*. Springer, 1998.

[8] X. Du and W. Chen, "Towards a better understanding of modeling feasibility robustness in engineering design," *ASME Journal of Mechanical Design*, 2000.

[9] M. Tichy, *Applied Methods of Structural Reliability*. Boston: Kluwer Academic Publishers, 1993.

[10] J. E. Stine *et al.*, "Freepdk: An open-source variation-aware design kit," in *ICMSE*, 2007.

[11] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *IEEE Trans. on Electron Devices*, vol. 53, no. 11, pp. 2816–2823, 2006.

[12] K. J. Kuhn, "Reducing variation in advanced logic technologies: Approaches to process and design for manufacturability of nanoscale cmos," in *IEDM*, 2007.

[13] K. J. Antreich and H. E. Graeb, "Circuit optimization driven by worst-case distances," in *ICCAD*, 1991.