Statistical Blockade: A Novel Method for Very Fast Monte Carlo Simulation of Rare Circuit Events, and its Application

Amith Singhee, Rob A. Rutenbar
Dept. of ECE, Carnegie Mellon University, Pittsburgh, Pennsylvania, 15213 USA
{asinghee,rutenbar}@ece.cmu.edu

Abstract

Circuit reliability under statistical process variation is an area of growing concern. For highly replicated circuits such as SRAMs and flip flops, a rare statistical event for one circuit may induce a not-so-rare system failure. Existing techniques perform poorly when tasked to generate both efficient sampling and sound statistics for these rare events. *Statistical Blockade* is a novel Monte Carlo technique that allows us to efficiently filter—to *block*—unwanted samples insufficiently rare in the tail distributions we seek. The method synthesizes ideas from data mining and Extreme Value Theory, and shows speedups of 10X -100X over standard Monte Carlo.

1. Introduction

Circuit reliability under statistical process variation is an area of growing concern. Designs that add excess safety margin, or rely on simplistic assumptions about "worst case" corners no longer suffice. Worse, for critical circuits such as SRAMs and flip flops, replicated across 10K - 10M instances on a large design, we have the new problem that statistically rare events are magnified by the sheer number of these elements. In such scenarios, an exceedingly rare event for one circuit may induce a not-so-rare failure for the entire system.

Monte Carlo analysis remains the gold standard for the required statistical modeling. Standard Monte Carlo techniques are, by construction, most efficient at sampling the statistically likely cases. Indeed, classical modifications such as *Importance Sampling* [1] allow Monte Carlo methods to avoid sampling these unlikely (i.e., unimportant) events. Ours is the mirror image problem: how can we efficiently sample *only* the statistically rare events? How can we *model* the statistics in the tails of these heavy-tailed distributions? Importance Sampling also gives us some help to sample in the tails [2], but changes the statistics of these rare samples. Unfortunately, we need both samples and rigorous statistics to determine the reliability of critical circuits like large SRAMs, or flips flop in aggressively clocked designs operating with small setup slack. Standard Monte Carlo methods are poorly suited to this important problem.

One avenue of attack is to abandon Monte Carlo. Several analytical and semi-analytical approaches have been suggested to model the behavior of SRAM cells [3][4][5] and digital circuits [6] in the presence of process variations. All suffer from approximations necessary to make the problem tractable. [4] and [6] assume a linear relationship between the statistical variables and the performance metrics (*e.g.* static noise margin), and assume that the statistical process parameters and resulting performance metrics are normally distributed. This can result in gross errors, especially while modeling rare events, as we shall show later. When the distribution varies significantly from Gaussian, [4] chooses an F-distribution in an *ad hoc* manner. [3] presents a complex analytical model limited to a specific transistor model (the transregional model) and further limited to only static noise margin analysis for the 6T SRAM cell. [5] again models only the static noise margin (SNM) for SRAM cells under assumptions of independence and identical distribution of the upper and lower SNM, which may not always be valid.

A different avenue of attack is to modify the Monte Carlo strategy. [2] shows how Importance Sampling can be used to predict failure probabilities. Recently, [7] applied an efficient formulation of these ideas for modeling rare failure events for single 6T SRAM cells, based on the concept of *Mixture Importance Sampling* from [8]. The approach uses real SPICE simulations with no approximating equations. However, the method only estimates the exceedence probability of a *single* value of the performance metric. A re-run is needed to obtain probability estimates for another value. No complete model of the tail of the distribution is computed. The method also combines all performance metrics to compute a failure probability, given *fixed* thresholds. Hence, there is no way to obtain separate probability estimates for each metric, other than a separate run per metric. Furthermore, given that [2] advises against importance sampling in high dimensions, it is unclear if this approach will scale efficiently to large circuits with many statistical parameters.

In this paper, we present a novel, general and efficient Monte Carlo method that addresses both problems previously described: very fast generation of samples—rare events—with sound models of the tail statistics for any performance metric. The method imposes almost no *a priori* limitations on the form of the statistics for the process parameters, device models, or performance metrics. The method is conceptually simple, and it exploits ideas from two rather nontraditional sources.

To obtain both samples and statistics for rare events, we may need to generate and evaluate an intractable number of Monte Carlo samples. G*enerating* each sample is neither challenging nor expensive: we are merely creating the parameters for a circuit. *Evaluating* the sample is expensive, because we simulate it. What if we could quickly *filter* these samples, and *block* those that are unlikely to fall in the low-probability tails we seek? Many samples could be generated, but very few simulated. We show how to exploit ideas from *data mining* [9] to build *classifier* structures, from a small set of Monte Carlo training samples, to create the necessary blocking filter. Given these samples, we show how to use the rigorous mathematics of *Extreme Value Theory* (EVT [10], the theory of the limiting behavior of sampled maxima and minima) to build sound models of these tail distributions. The essential "blocking" activity of the filter gives the technique its name: *Statistical Blockade*.

Statistical blockade has been tested on both SRAM and flip-flop designs, including a complete 64-cell SRAM column (a 403-parameter problem), accounting for both local and global variations. (In contrast to several prior studies [5-6,9] we shall see that simulating only one cell does not correctly estimate the critical tail statistics.) However, statistical blockade allows us to generate both samples and accurate statistics, with speedups of **10X -100X** over standard Monte Carlo.

978-3-9810801-2-4/DATE07 © 2007 EDAA

This paper is organized as follows. Section 2 reviews the mathematics for modeling rare event tail distributions, derived from EVT. Section 3 develops our tail distribution fitting strategy, based on probability-weighted moments, and our method for probability prediction, once the model has been built. Section 4 develops the core of the statistical blockade method: an efficient tail sampling strategy, using a classifier-based blocking filter. Section 5 presents experimental results. Section 6 offers concluding remarks.

2. Extreme Value Theory

EVT provides us with mathematical tools to build models of the tails of distributions. It has been used extensively in climatology and risk management, among other applications: wherever the probability of extreme and rare events needs to be modeled. Here we introduce the mathematical concepts from EVT that our approach relies on.

Suppose we define a threshold t for some random variable (e.g. the SNM of an SRAM cell) with cumulative distribution (CDF) $F(x)$: All values above t constitute the *tail* of the distribution. Throughout this paper, we are considering only the upper tail: this is without loss of generality, since a simple sign change converts a lower tail to the upper tail. Now define the conditional CDF of excesses above t as

$$F_t(x) = P\{X - t \geq x | X \geq t\} = \frac{F(x+t) - F(t)}{1 - F(t)} \text{ for } x \geq 0 \quad (1)$$

An important distribution in the theory of extreme values is the *Generalized Pareto Distribution* (GPD), which has the following CDF:

$$G_{a,k}(x) = \begin{cases} 1 - (1 + kx/a)^{1/k} & , k \neq 0 \\ 1 - e^{-x/a} & , k = 0 \end{cases} \quad (2)$$

The seminal result we exploit is from Balkema and de Haan [11] and Pickands [12] (referred to as *BdP*) who proved that

$$\lim_{t \to \infty} \sup_{x \geq 0} |F_t(x) - G_{a,k}(x)| = 0 \quad (3)$$

if and only if F is in the *maximum domain of attraction* (MDA) of the *Generalized Extreme Value* distribution (GEV): $F \in MDA(H_\eta)$. This means that when the distribution F satisfies the given condition ($F \in MDA(H_\eta)$), the conditional CDF of F tends, as we move the threshold farther and farther out on the tail, towards a particularly tractable analytical form. Let us look at the condition in more detail.

The GEV CDF is as follows

$$H_\eta(x) = \begin{cases} e^{-(1+\eta x)^{-1/\eta}} & , \eta \neq 0 \\ e^{-e^{-x}} & , \eta = 0 \end{cases} \text{ where } 1 + \eta x > 0 \quad (4)$$

It combines three simpler distributions into one unified form:

- for $\eta = 0$ we get the *Gumbel-type* (or Type I) distribution

$$\Lambda(x) = e^{-e^{-x}} \quad (5)$$

- for $\eta > 0$, we get the *Fréchet-type* (or Type II) distribution

$$\Theta_\alpha(x) = e^{-x^{-\alpha}} \text{ for } x > 0 \quad (6)$$

- for $\eta < 0$, we get the *Weibull-type* (or Type III) distribution

$$\Psi_\alpha(x) = e^{-|x|^\alpha} \text{ for } x < 0 \quad (7)$$

Let us now look at what the "maximum domain of attraction" means. Consider the maxima (M_n) of n i.i.d. random variables. Suppose there exist normalizing constants a_n and b_n, such that

$$P\{(M_n - b_n)/a_n \leq x\} = F^n(a_n x + b_n) \to H(x) \text{ as } n \to \infty \quad (8)$$

for some non-degenerate $H(x)$. Then we say that F is "in the maximum domain of attraction" of H. In other words, the maxima of n i.i.d. random variables with CDF F, when properly normalized, converge in distribution to a random variable with the distribution H. Fisher and Tippett [13] and Gnedenko [14] showed that for a large class of distributions,

$$F \in MDA(H) \Rightarrow H \text{ is of type } H_\eta \quad (9)$$

For example [10], MDA(Λ) includes the normal, exponential, gamma and lognormal distributions; MDA(Θ_α) includes the Pareto, Burr, log-gamma, Cauchy and t-distributions; MDA(Ψ_α) includes finite-tailed distributions like the uniform and beta distributions. Hence, for a large class of distributions, the *BdP* theorem holds true. In other words, if we can generate enough points in the tail of a distribution ($x \geq t$), in most cases, we can fit a GPD to the data and make predictions further out in the tail.

This is a remarkably practical and useful result for the rare circuit event scenarios we seek to model. In particular, it shows that most prior *ad hoc* fitting strategies are at best sub-optimal, and at worst, simply wrong. Let us next consider how to use these results.

3. Model Fitting and Prediction

Assuming we can generate points in the tail, there remains the problem of fitting a GPD form to the conditional CDF. Several options are available here [15]: moment matching, maximum likelihood estimation (MLE) and probability weighted moments (PWM) [16]. We have chosen PWM because it seems to have lower bias [15] and does not have the convergence problems of MLE.

The PWMs of a continuous random variable x with CDF F are the quantities

$$M_{p,r,s} = E[x^p \{F(x)\}^r \{1 - F(x)\}^s] \quad (10)$$

which often have simpler relationships with the distribution parameters than conventional moments $M_{p,0,0}$. For the GPD it is convenient to use these particular PWMs

$$\alpha_s = M_{1,0,s} = E[x\{1 - F(x)\}^s] = \frac{a}{(s+1)(s+1+k)} \quad (11)$$

which exist for $k > -1$: this is true for most cases of interest [15]. The GPD parameters are then given by

$$a = \frac{2\alpha_0\alpha_1}{\alpha_0 - 2\alpha_1}, \; k = \frac{\alpha_0}{\alpha_0 - 2\alpha_1} - 2 \quad (12)$$

where the PWMs are estimated from the samples as

$$\tilde{\alpha}_i = n^{-1} \sum_{j=1}^{n} (1 - p_{j|n})^i x_{j|n} \quad (13)$$

where $x_{1|n} \leq \ldots \leq x_{n|n}$ are the ordered samples and $p_{j|n} = (j + \gamma)/(n + \delta)$. $\gamma = -0.35$ and $\delta = 0$ are as suggested in [15].

Given the ability to fit the GPD form, now let us consider the problem of predicting useful probabilities. Once we have a GPD model of the conditional CDF above a threshold t, we can predict the exceedence probability—the *failure* probability—for any value x_f:

$$P(X > x_f) = [1 - P(X \leq t)][1 - F_t(x_f - t)] \quad (14)$$

Here, $P(X \le t)$ can be computed using empirical data obtained from standard Monte Carlo, or more sophisticated variance reduction techniques, for example, mixture importance sampling [7]. $F_t(x_f - t)$ is just the prediction by the GPD model. Hence, we can write (14) as

$$P(X > x_f) = [1 - F(t)][1 - G_{a,k}(x_f - t)] \quad (15)$$

4. Statistical Blockade: Classification-based Sampling

Even with all the useful theory presented above, we still need a way to efficiently generate samples in the tail of the distribution of the performance metric of a circuit. Standard Monte Carlo (MC) is very unsuited to this job, because it generates samples that follow the complete distribution. The problem is severe for rare event statistics: if our threshold t is the 99% point of the distribution, only one out of 100 simulations will be useful for building the tail model.

Our approach is to build a so-called *classifier* to filter out candidate MC points that will not generate a performance value in the tail. Then, we simulate only those MC points that will generate points in the tail. For clarity, we shall refer to this structure as the *blockade filter*, and its action as *blockade filtering*. We borrow ideas from the data mining community [9] to build the filter. A *classifier* is an indicator function that allows us to determine set membership for complex, high-dimensional, nonlinear data. Given a data point, the classifier reports true or false on the membership of this point in some arbitrary set. For statistical blockade, this is the set of parameter values *not* in the extremes of the distributional tail we seek. The classifier is built from a relatively small set of representative sample data, and as we shall see, need not be perfectly accurate to be effective.

Let us look at this filter, and its construction. Suppose the statistical parameters (V_t, t_{ox}, etc.) in a circuit are denoted by s_i, and the performance metric being measured is y. Our sampling strategy tries to simulate only those points $\{s_i\}$, that result in values of $y \ge t$. This is accomplished in three steps (shown in Fig. 1):

1) *Perform initial sampling* to generate data to build a classifier. This initial sampling is also used for estimating $F(t)$ in (15), and could be standard Monte Carlo or importance sampling.
2) *Build a classifier* using a classification threshold t_c. To minimize false negatives (tail points classified as non-tail points), choose $t_c < t$.
3) *Generate more samples* using MC, following the CDF F, but simulate only those that are classified as tail points.

Using the tail points generated by the blockade-filtered sampling, we can then build a conditional CDF model for the tail, using the tools of Sections 2 and 3. As long as the number of false negatives is acceptably low, the simulated tail points are true to the actual distribution. Hence, there is no need to unbias the estimates. Note that the approach is reminiscent of acceptance-rejection sampling [1].

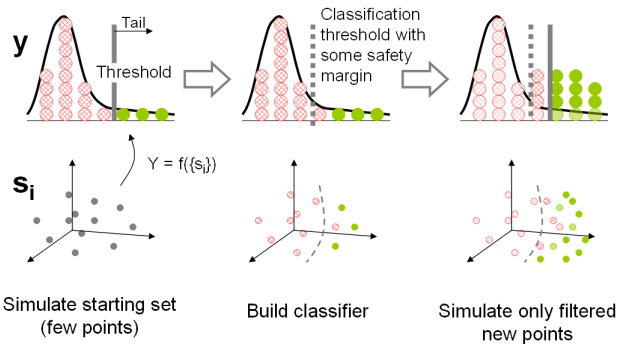

FIGURE 1. Classification based sampling

In this work, the classifier used is a *Support Vector Machine* (SVM, [17]). The time for model building and classification is negligible compared to the total simulation time. Apart from this practical consideration, there is no restriction on the type of classifier that can be used. Classification is a rich and active field of research in the data mining community and there are many options for choosing a classifier [9]. SVMs are a popular, well researched classifier strategy, and optimized implementations are readily available [17].

5. Experimental Results

We now apply the statistical blockade method to three testcases: a single 90nm SRAM cell, a 45nm master-slave flip-flop and a full 64-bit 90nm SRAM column. The initial sampling to construct each blockade filter was a standard MC run of 1000 points. An SVM classifier was built using the 97% point (of each relevant performance metric) as the classification threshold t_c. The tail threshold t was defined as the 99% point.

One technical point to note about the SVM construction: since the sample set is biased with many more points in the body of the distribution than in the tail, we need to unbias the classification error [18]. Suppose that, of the 1000 simulated training points, $T \ll 1000$ actually fall into the tail we seek. Since the two classification sets (true/false) have an unbalanced number of points, the SVM classifier will be biased toward the body (1000-T points). Even if all T of the tail points are misclassified, the error rate is quite low if the body is classified correctly. Hence, classification error in the tail is penalized more—by a weighting factor of roughly T—than errors in the body, to try to avoid missing tail points. We use a weight value of 30 for these results.

5.1 Single 6-T SRAM Cell

The first testcase is shown in Fig. 2: a 6-T SRAM cell, with bit-lines connected to a column multiplexor and a non-restoring write driver. The metric being measured is the write time τ_w: the time between the wordline going high to the non-driven cell node (node 2) transitioning. Here, "going high" and "transitioning" imply crossing 50% of the full voltage change. The device models used are from the Cadence 90nm Generic PDK library. There are 9 statistical parameters: 8 V_t variations to model random dopant fluctuation (RDF, [19]) effects in the transistors named in the figure, and 1 global gate-oxide variation. All variations are assumed to be normally distributed about the nominal value. The V_t standard deviation is

$$\sigma(V_t) = \frac{5mV}{\sqrt{WL}} \text{ where W,L are in } \mu\text{m} \quad (16)$$

This variation is too large for the 90nm process, but is in the expected range for more scaled technologies; this creates a good stress test for the method. The gate-oxide standard deviation is taken as 2%.

100,000 MC points were blockade-filtered through the classifier, generating 4,379 tail candidates. After simulating these 4,379 points, 978 "true" tail points were obtained. The tail model obtained from these points is compared with the empirical tail conditional CDF obtained after simulating 1 million MC points, in Fig. 3. Table 1 shows a comparison of the failure probability predictions for different values of τ_w, expressed as equivalent sigma points:

$$x_\sigma = \Phi^{-1}(1 - P(X > x_f)) \quad (17)$$

where Φ is the standard normal CDF. This is the equivalent point on a standard normal that would have the same cumulative probability. For example, $x_\sigma = 3$ implies a cumulative probability of 0.99865 and

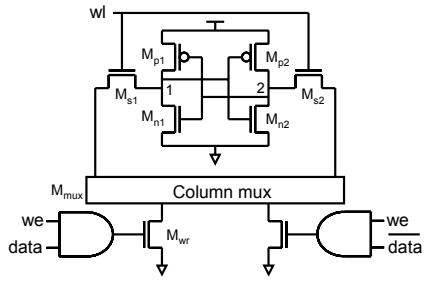

FIGURE 2. 6-T SRAM cell with column mux and write drivers. V_t variation on named devices and global t_{ox} variation.

.

τ_w	**Standard MC** (1M sims)	**GPD *No* Blockade Filter** (1M sims)	**GPD *With* Blockade Filter** (5,379 sims)
2.4	3.404	3.408	3.379
2.5	3.886	3.886	3.868
2.6	4.526	4.354	4.352
2.7	∞	4.821	4.845
2.8	∞	5.297	5.356
2.9	∞	5.789	5.899
3.0	∞	6.310	6.493

TABLE 1. Comparison of predictions by Monte Carlo, Monte Carlo with tail modeling and statistical blockade filtering, for single SRAM cell. The number of simulations includes the 1000 training samples.

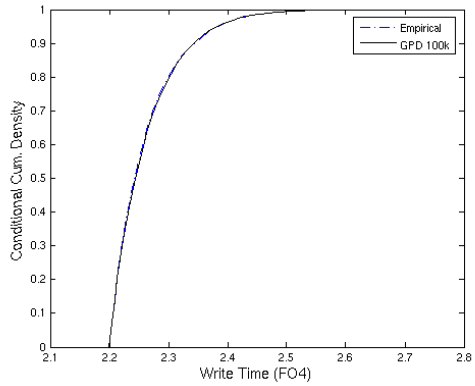

FIGURE 3. Comparison of tail model CDF (5379 simulations) with empirical tail CDF (1 million simulations)

a failure probability of 0.00135. The delays are expressed as multiples of the fanout-of-four (FO4) delay of the process. The table also shows x_σ predictions from an accurate tail model built using the 1 million MC points, without any filtering. The empirical prediction fails beyond 2.7 FO4 because there are simply *no* points generated by the MC run so far out in the tail (beyond 4.8σ).

The table shows two important advantages of our approach:

- Even without any filtering, the GPD tail model is better than Monte Carlo, since it can be used to predict probabilities far out in the tail, even when there are no points that far out.
- Using blockade filtering, coupled with the tail model, we can drastically reduce the number of simulations (from 1 million to 5,379) and still generate a reliable tail model.

5.2 Master-Slave Flip-Flop with Scan Chain

A large chip can have tens of thousands of instances of the same flip-flop. Typically, these flip-flops are in a scan chain to enable

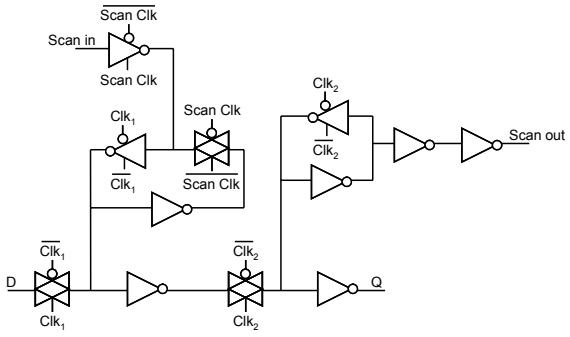

FIGURE 4. Master-Slave Flip-Flop with Scan Chain

rigorous testing. Random threshold variation in the scan chain transistors can also impact the performance of the flip-flop. Hence, our next testcase (Fig. 4) is a commonly seen Master-Slave Flip-Flop with scan chain (MSFF).

The design has been implemented using the 45 nm CMOS Predictive Technology Models from [20]. Variations considered include RDF for all transistors in the circuit and one global gate-oxide variation. Threshold variation is modeled as normally distributed V_t variation:

$$\sigma(V_t) = 0.0135\frac{V_{t0}}{\sqrt{WL}} \text{ where W,L are in } \mu\text{m} \qquad (18)$$

V_{t0} is the nominal threshold voltage. This results in 30% standard deviation for a minimum-sized transistor. The t_{ox} standard deviation is taken as 2%. The metric being measured is the clock-output delay, τ_{cq} in terms of the FO4 delay. A GPD model was built using 692 true tail points, obtained from 7,785 candidates blockade filtered from 100,000 MC samples. Fig. 5 compares this model with (1) the empirical CDF from 500,000 standard MC simulations, and (2) a GPD model built from after blockade filtering these 500,000 points**.** The discrepancy of the models can be explained by looking at the empirical PDF of the delay in Fig. 6. Due to the heavy tail, slight variations in the tail samples chosen can cause large variations in the model. Our method is still able to generate an acceptably accurate model, as is evident by the comparison of x_σ in Table 2. Standard MC starts under-estimating the failure probability (over-estimating x_σ) far out in the tail (from row 3 on). The tail model has much better predictive power (column 2): x_σ = 4.283 implies a failure probability of 9.2 ppm. Even with blockade filtering, the tail model is still quite accurate. The table also shows the estimates from a standard Gaussian distribution fit to 20,000 MC

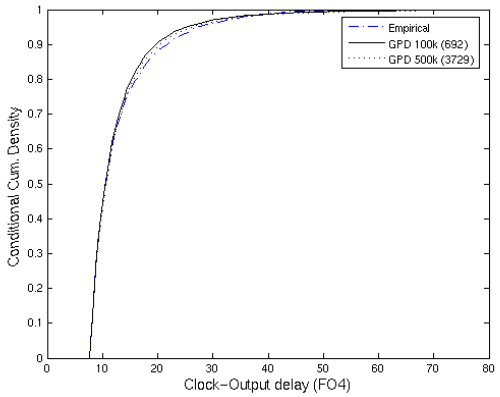

FIGURE 5. Tail model for MSFF (1692 and 4729 simulations) compared with empirical model (500,000 simulations)

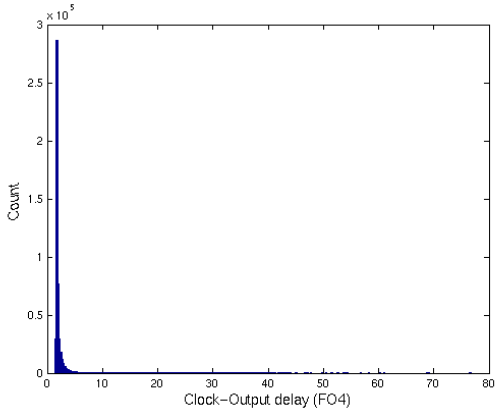

FIGURE 6. Probability density plot for Clock-Output delay of the MSFF, showing a long, heavy tail.

τ_{cq}	Standard MC (500K sims)	GPD *No* Blockade Filter (500K sims)	GPD *With* Blockade Filter (8,785 sims)	Gaussian Tail Approx (20K sims)
30	3.424	3.466	3.431	22.127
40	3.724	3.686	3.661	30.05
50	4.008	3.854	3.837	37.974
60	4.219	3.990	3.978	45.898
70	4.607	4.102	4.095	53.821
80	∞	4.199	4.195	61.745
90	∞	4.283	4.282	69.669

TABLE 2. Comparison of predictions by MC, MC with GPD modeling, blockade filtered GPD modeling, and standard Gaussian approximation, for MSFF. The number of simulations includes the 1000 training samples.

points: it is obvious that such a simplifying assumption *severely* under-estimates the failure probability.

5.3 64-bit SRAM Column

The next testcase is a 64-bit SRAM column, with non-restoring write driver and column multiplexor (Fig. 7).Only one cell is being accessed, while all the other wordlines are turned off. Random threshold variation on all 402 devices (including the write driver and column mux) are considered, along with a global gate-oxide variation. The device and variation models are the same 90nm technology as the single SRAM cell (Section 5.1). In scaled technologies, leakage is no longer negligible. Hence, process variations on devices that are meant to be inaccessible can also impact the overall behavior of a circuit. This testcase allows us to see the impact of leakage through the 63 off cells, along with variations in the write driver.

The metric measured is the write time (τ_w), from wl_0 to node 2. The number of statistical parameters is 403 in this case. Building a classifier with only 1000 points in 403 dimensional space is nearly impossible. Hence, the dimensionality is reduced by choosing only those parameters that significantly affect the output. We employ standard statistical sensitivity techniques. We measure this significance with Spearman's Rank Correlation Coefficient [21], r_s. Suppose R_i and S_i are the ranks of corresponding values of two variables in a dataset, then their rank correlation is given as:

$$r_s = \frac{\sum_i (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i (R_i - \bar{R})^2}\sqrt{\sum_i (S_i - \bar{S})^2}} \quad (19)$$

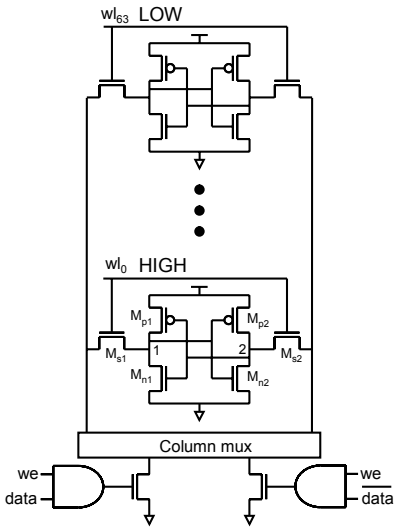

FIGURE 7. 64-bit SRAM column with column mux and write drivers. V_t variation on all devices and global t_{ox} variation.

This measure of correlation is more robust than a linear Pearson's correlation, in the presence of non-linear relationships in the data. Fig. 8 shows the sorted magnitudes of the 403 rank correlation values, computed between the statistical parameters and the output τ_w. For classification, only the parameters with $|r_s| > 0.1$ were chosen. This reduced the dimensionality to only 11: the devices chosen by this method were the pull-down and output devices in the active AND gate, the column mux device, the bitline pull-down devices and all devices in the 6-T cell, except for M_{p2} (since node 2 is being pulled down in this case). This selection coincides with a designer's intuition of the devices that would have the most impact on the write time in this testcase.

The empirical CDF from 100,000 MC samples is compared with the tail model obtained by blockade filtering 20,000 MC samples (218 true tail points from 1046 filtered candidates) in Fig. 9. Also shown is the tail model obtained by blockade filtering the 100,000 MC samples. Table 3 compares the following: the x_σ predictions from standard MC; a GPD tail model with no filtering; two *different* GPD tail models with filtering of 20,000 and 100,000 points, respectively; and a standard Gaussian fit to 20,000 points. We can see that the 218 true tail points obtained by blockade filtering only

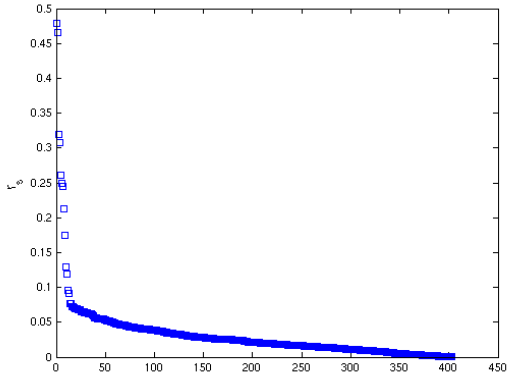

FIGURE 8. Absolute values of rank correlation between the statistical parameters and write time of the SRAM column.

τ_w	**Standard MC** (100K sims)	**GPD *No* Blockade Filter** (100K sims)	**GPD *w/* Blockade Filter** (20K pts 2,046 sims)	**GPD *w/* Blockade Filter** (100K pts 6,314 sims)	**Standard Gaussian Approx.** (20K sims)
2.7	2.966	2.986	2.990	3.010	3.364
2.8	3.367	3.373	3.425	3.390	3.898
2.9	3.808	3.743	3.900	3.747	4.432
3.0	∞	4.101	4.448	4.088	4.966
3.1	∞	4.452	5.138	4.416	5.499
3.2	∞	4.799	6.180	4.736	6.033
3.3	∞	5.147	-	5.049	6.567
3.4	∞	5.496	-	5.357	7.100

TABLE 3. Comparison of predictions by MC, MC with tail modeling, blockade filtered tail modeling and Gaussian approximation, for SRAM column. The number of simulations includes the 1000 training samples.

20,000 MC samples is not enough to build a reliable tail model. However, we get much better results using the 1077 true tail points obtained by blockade filtering 100,000 MC samples (5314 simulations). The Gaussian again under-estimates the failure probability.

Comparing with Table 1, we can see that simulating variations in a single cell, without modeling variation in the environment circuitry (other cells in the column and the write driver itself), can lead to large underestimation of the delay spread: 3.0 FO4 delay is estimated as a 6.3σ point (Table 1), while it is actually a 4.1σ point (Table 3).

Before concluding, we mention two points. First, across all three testcases, we see significant improvements in accuracy over simple Gaussian fits, and similar improvements in fitting if we use our GPD model and simple MC sampling. However, we also see significant speedups over simple Monte Carlo, ranging from roughly one to two orders of magnitude.

Finally, we mention an obvious extension to these ideas. The testcases shown herein all measure a single performance metric. Our method is, however, flexible enough to accommodate multiple metrics: multiple classifiers can be trained from the same training set, one for each metric. Each classifier would then identify potential tail points for its corresponding metric, which can be simulated and used to build a tail model for every metric. In the worst case, the tail samples of two metrics might be mutually exclusive, resulting in approximately twice the number of simulations as compared to the case of a single metric. In the best case, the tail samples of the metrics would overlap and there would not be any significant increase in the number of simulations.

6. Conclusions

Statistical blockade is a novel, efficient and flexible framework for (1) generating samples in the tails of distributions of circuit performance metrics, and (2) deriving sound statistical models of these tails. This enables us to make predictions of failure probabilities given thresholds far out in the tails. This capability has become critical for reliable and efficient design of high-replication circuits, such as SRAMs, as transistor sizes move deeply into the nanometer regime. Our methods offer both significantly higher accuracy than standard Monte Carlo, and speedups of one to two orders of magnitude across a range of realistic circuit testcases and variations.

Acknowledgements: this work was supported by the MARCO/ DARPA Focus Research Center for Circuit and System Solutions (C2S2) and the Semiconductor Research Corp.

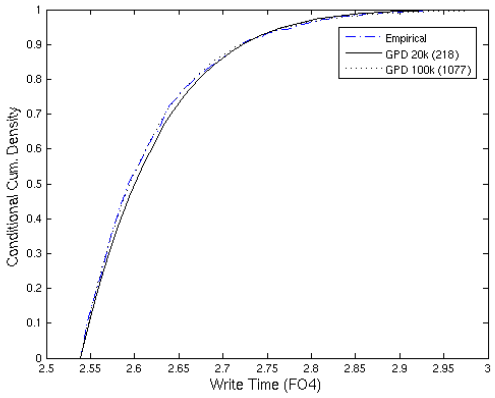

FIGURE 9. Tail model for SRAM column (2046 and 6314 simulations) compared with empirical model (100,000 simulations)

References

[1] G.S. Fishman, "A First Course in Monte Carlo", Duxbury Press, Oct. 2005.

[2] D.E. Hocevar, M.R. Lightner, T.N. Trick, "A Study of Variance Reduction Techniques for Estimating Circuit Yields', *IEEE Trans. CAD*, 2(3), July, 1983.

[3] A.J. Bhavnagarwala, X. Tang, J.D. Meindl, "The Impact of Intrinsic Device Fluctuations on CMOS SRAM Cell Stability", *J.Solid State Circuits*, 26(4), pp 658-665, Apr. 2001.

[4] S. Mukhopadhyay, H. Mahmoodi, K. Roy, "Statistical Design and Optimization of SRAM Cell for Yield Enhancement", *Proc. ICCAD*, 2004.

[5] B.H. Calhoun, A. Chandrakasan, "Analyzing Static Noise Margin for Sub-threshold SRAM in 65nm CMOS", *Proc. ESSCIRC*, 2005.

[6] H. Mahmoodi, S. Mukhopadhyay, K. Roy, "Estimation of Delay Variations due to Random-Dopant Fluctuations in Nanoscale CMOS Circuits", *J. Solid State Circuits*, 40(3), pp 1787-1796, Sep. 2005.

[7] R. Kanj, R. Joshi, S. Nassif, "Mixture Importance Sampling and its Application to the Analysis of SRAM Designs in the Presence of Rare Failure Events", *Proc. DAC*, 2006.

[8] T.C. Hesterberg, "Advances in Importance Sampling", PhD Dissertation, Dept. of Statistics, Stanford University, 1988, 2003.

[9] T. Hastie, R. Tibshirani, J. Friedman, "The Elements of Statistical Learning", Springer Verlag, 2003.

[10] A.J. McNeil, "Estimating the Tails of Loss Severity Distributions using Extreme Value Theory", *ASTIN Bulletin*, 27(1), pp 117-137, 1997.

[11] A. Balkema, L. de Haan, "Residual life time at great age", *Annals of Probability*, 2(5), pp 792-804, 1974

[12] J. Pickands III, "Statistical Inference Using Extreme Order Statistics", *The Annals of Statistics*, 3(1), pp 119-131, Jan. 1975.

[13] R. Fisher, L. Tippett, "Limiting Forms of the Frequency Distribution of the Largest or Smallest Member of a Sample", *Proc. Cambridge Phil. Soc.*, 24, pp 180-190, 1928.

[14] B. Gnedenko, "Sur La Distribution Limite Du Terme Maximum D'Une Serie Aleatoire", *The Annals of Mathematics*, 44(3), Jul. 1943.

[15] J.R.M. Hosking, J.R. Wallis, "Parameter and Quantile Estimation for the Generalized Pareto Distribution", *Technometrics*, 29(3), pp 339-349, Aug. 1987

[16] J.A. Greenwood, J.M. Landwehr, N.C. Matalas, J.R. Wallis, "Probability Weighted Moments: Definition and Relation to Parameters of Several Distributions Expressable in Inverse Form", *Water Resources Research*, 15, pp 1049-1054, 1979.

[17] T. Joachims, "Making Large-Scale SVM Learning Practical", LS8-Report, 24, Universität Dortmund, 1998.

[18] K. Morik, P. Brockhausen, T. Joachims, "Combining Statistical Learning with a Knowledge-based Approach - A Case Study in Intensive Care Monitoring", *Proc. 16th Int'l Conf. on Machine Learning*, 1999.

[19] D.J. Frank, Y. Taur, M. Ieong, H.P. Wong, "Monte Carlo Modeling of Threshold Variation due to Dopant Fluctuations", *Symp. VLSI Technology*, 1999.

[20] http://www.eas.asu.edu/~ptm/

[21] G.E. Noether, "Introduction to Statistics: The Nonparametric Way", Springer, 1990.