SSFB: A Highly-Efficient and Scalable Simulation Reduction Technique for SRAM Yield Analysis

Manish Rana and Ramon Canal
Department d'Arquitectura de Computadors,
Universitat Politècnica de Catalunya,
Barcelona, Spain
{mrana,rcanal}@ac.upc.edu

***Abstract*—Estimating extremely low SRAM failure-probabilities by conventional Monte Carlo (MC) approach requires hundreds-of-thousands simulations making it an impractical approach. To alleviate this problem, failure-probability estimation methods with a smaller number of simulations have recently been proposed, most notably variants of consecutive mean-shift based Importance Sampling (IS). In this method, a large amount of time is spent simulating data points that will eventually be discarded in favor of other data-points with minimum norm. This can potentially increase the simulation time by orders of magnitude. To solve this very important limitation, in this paper, we introduce SSFB: a novel SRAM failure-probability estimation method that has much better cognizance of the data points compared to conventional approaches. The proposed method starts with radial simulation of a single point and reduces discarded simulations by: a) random sampling -only- when it reaches a failure boundary and after that continues again with radial simulation of a chosen point, and b) random sampling is performed -only- within a specific failure-range which decreases in each iteration. The proposed method is also scalable to higher dimensions (more input variables) as sampling is done on the surface of the hyper-sphere, rather than within-the-hypersphere as other techniques do. Our results show that using our method we can achieve an overall 40x reduction in simulations compared to consecutive mean-shift IS methods while remaining within the 0.01-Sigma accuracy.**

I. Introduction

Process variations are the most important concern for circuit designer these days. The influence of random variations on the circuit performance is increasing with progressive technology scaling. SRAM arrays designed with minimum feature size are especially affected by the inevitable presence of these random variations, resulting in millions of dollars wasted because of poor yield. Variability aware circuit-design requires performing a huge number of simulations, thereby wasting the computing resources and time. A breakthrough is desperately needed that enables variation-resistant circuit-design with a smaller burden of long simulations. This paper addresses this very important and highly relevant problem for the present-day circuit designers.

Estimating the robustness of the whole SRAM memory array requires estimating the failure-probabilities of a single bit-cell in the array. For a failure-rate of 0.1% for 10MB SRAM array, the bit-cell failure rate is required to be below 10^{-10}. This means more than 10^{10} simulations are needed to obtain a single failure instance when using the Monte-Carlo method. This many simulations translates to a large expense of computing time and power. Some methods have recently been proposed that estimate failure-probabilities in a smaller amount of simulations than Monte-Carlo. A large class of these proposals use Importance-Sampling(IS) technique. The accuracy of IS method is heavily dependent on the shifted distribution used which should be as close as possible to the failure distribution. However since the failure distribution is unknown before the simulation, methods like Mixture-Importance Sampling[7] use a combination of different distributions. Other methods simplify the approach by performing IS with a mean-shifted distribution at the Most-Probable-Failure-Point(MPFP). Importance Sampling augmented with MPFP has been explored in [4],[6] and [9]. However, the existing methods of finding MPFP are not very efficient when considering the simulations required. As an example, the consecutive-mean shift method[9] in each iteration performs a) random sampling from a mean-shifted distribution, b) finds the minimum-norm point from the sampled points and shifts the mean of sampling distribution to this minimum-norm point. This is inefficient because in each iteration, of all the points randomly sampled, only one point is kept and every other point is discarded with no addition to knowledge of the failure region for the next iteration.

In this paper, we propose that, in each iteration, we compute a rough estimate of the failure-range in each dimension rather than the single minimum-norm point. This saves simulation time by not wasting computing resources on simulations that will be eventually discarded. The SSFB method has its foundations on the observation that the MPFP has to be on the failure boundary. As such the region of interest for sampling is the failure boundary. Sample points that do not lay on the failure boundary do not contribute anything to the objective of finding the MPFP. The underlying idea of SSFB is that instead of repeatedly random sampling in each iteration for minimum-norm point, we start with radial simulation of just a single point and only when that point reaches the failure boundary it performs random sampling. Its easy to identify when a point reaches the failure boundary as its next radial simulation will not result in a failure. The spherical-surface random sampling is then performed at this failure-boundary point to find a rough estimate of failure-ranges in each dimension. The failure-

978-3-9815370-2-4/DATE14/ ©2014 EDAA

region with the largest failure-range will most likely contain the MPFP and so, the middle point of the largest failure-range is chosen as the simulation-point for the subsequent iterations.

This papers makes following contributions:

- A novel method SSFB for estimating MPFP which has three major distinctions from existing methods-
  - Efficient use of the surface-spherical sampling as compared to the within-the-hypersphere sampling used in the existing methods. With this shift in sampling approach, high simulation density can be achieved in fewer number of simulations on the hypersphere-surface than with within-the-hypersphere sampling.
  - Finding failure ranges in each dimension and restricting sampling only within this failure range. First, this allows it to gain better knowledge of the failure region than finding only the minimum-norm point in each iteration. Second, it saves simulations by not simulating non-failure points.
  - Reduce the number of random samplings. This is achieved by the radial simulation of only one point till it reaches failure boundary and, only then, it performs the random sampling.
    - Its only because of the cumulative effect of the last two factors that spherical surface sampling is able to perform better than within-the-hypersphere sampling by requiring overall less number of simulations.
- Application of the method to 6T SRAM read-failure-probability estimation and its comparison with existing methods showing 40x reduction in simulation with no loss in accuracy.

The rest of this paper is organized as follows. In section II, background for this work is explained along with a brief description of existing MPFP based IS methods. In section III, we present the SSFB method. In section IV, SSFB is compared with the existing methods by analyzing SRAM read-failure-probability. Finally, we conclude the paper in section V.

II. Background

A. Monte-Carlo and Importance Sampling

The analytical methods become intractable for large dimensions as they involve solving high-dimensional integrals. Monte-Carlo analysis is suitable for those high-dimensional integration problems that are too complex to be solved using the analytical and/or numerical methods. It gives a non-biased accurate estimate with a smaller number of simulations as compared to the analytical methods.

Mathematically Monte-Carlo is described as follows:

Consider a random variable X with probability density function $f(x)$. Then the expected value of the function $g(x)$ over X is defined as : $E[g(x)] = \int g(x)f(x)dx$.

Computing this integral with analytical methods may be cumbersome for higher dimensions. For these types of integrals, Monte-Carlo provides an estimator for the function by performing repeated simulations. To do so, n number of samples $< x1, x2, x3..xn >$ are taken from X and mean of $g(x)$ is computed over these sampled values. Thus the Monte-Carlo estimator function is defined as : $\bar{}g(x) = \frac{1}{n}\sum g(x)$.

By the weak law of large numbers we have for arbitrarily small ϵ, $lim_{n\dashrightarrow\infty}P(|\bar{g}(x) - E[g(x)]| \geqslant \epsilon) = 0$, which says that for large n samples, the probability of deviation between $\bar{g}(x)$ and $E[g(x)]$ becomes negligible. Thus, higher simulation efficiency of Monte-Carlo requires number of simulations n to be large. The number of samples required to obtain $(1 - \epsilon)100\%$ accuracy at $(1 - \delta)100\%$ confidence are given by : $N_{\epsilon,\delta} = (log_{10}(1/\delta))/p\epsilon^2$.

To estimate failure-probability using Monte-Carlo analysis, an indicator function $I(x)$ is needed which is defined as $I(x) = 0,\ if\, x \in A,\ else\, I(x) = 1$, where A is the failure set. With this indicator function the estimator for failure-probability is defined as : $P(x \in A) = E(I_{\{A\}}(x))$.

Importance sampling is a method of reducing the variance of the Monte-Carlo estimator based on the idea of sampling from an alternative distribution $h(x)$ with more failure samples, instead of the original distribution $g(x)$. Sampling from this distribution will give more failure points and a reliable estimate of failure-probability can be found with a smaller number of simulations. The failure-probability estimator using importance sampling is given below:

$$P_{IS} = \frac{1}{N}\sum I_{\{A\}}(x).\frac{h(x)}{g(x)} \tag{1}$$

To minimize the variance of this estimator, the new sampling distribution h(x) should be as close as possible to the original distribution g(x).

B. Importance Sampling (IS) based Probability Failure

The accuracy of the failure-probability estimator is dependent on this alternate sampling distribution which ideally should be similar to the original distribution. The Most-Probable-Failure-Point (MPFP) lies on the failure boundary and is also the closest failure point to the origin. Hence by shifting the mean of the original distribution to MPFP point we obtain an alternate distribution that is close to original distribution and provides more failure points. Next, we describe briefly three methods that find the MPFP point using random sampling for subsequent use in the IS estimator for probability failure.

1) IHS-DHS [8]: The first method starts with Incremental-Hypersphere-Sampling(IHS) where the failure samples are searched in a region within two hyper-spheres. If none of the samples result in failure then the radii of two hyperspheres is increased by 1σ. Once a failure sample is found, the IHS is followed by Decremental-Hypersphere-Sampling(DHS) in which the radii of the hyperspheres is reduced in steps of 0.1σ. In this step, the important quadrants for finding failure samples are identified and the subsequent sampling is restricted these important quadrants only. The DHS continues in the important quadrants till the minimum-norm point is found. Since in this method simulations have to cover the 0.1σ width during DHS and 1σ during IHS, more simulations are needed; otherwise, as the simulation sampling density is not big enough, we obtain only a rough estimate of the MPFP point.

2) Mean-shift[9]: The second method overcomes this problem of limited simulation sampling density by consecutively shifting the mean of the hyperspheres with smaller σ. It consists of three steps:

[a]Sampling form extended hyper-sphere

The radius of the Hypersphere is increased in steps of 1σ till a failure sample is found. If more than one sample points fail, then the minimum-norm sample points is chosen as MPFP-1.

[b]Sampling from mean-Shifted Hyper-sphere

The mean of the hypersphere is shifted to MPFP-1 in this step and sampling with original sigma is performed to find the failure points. Since MPFP-1 lies close to the failure boundary, the actual MPFP will be close to MPFP-1. As such minimum-norm point from the samples is chosen as MPFP-2.

[c]Sampling from Consecutively Mean-shifted IS

The mean of hypersphere is shifted to MPFP-2. Now that the actual MPFP lies nearby the mean of hypersphere, the sigma for sampling is reduced by $\frac{1}{3}$ to increase the simulation density. An iterative sampling and shifting of mean is performed to find minimum-norm point till the means of two consecutive hyperspheres are within a distance of 0.01σ, hence giving a more accurate estimate than the first method.

3) Seq-IS[3]: This method is based on the idea of using Bayesian-Filters to predict a better failure distribution based on the current failure distribution results. In a particle filter, a fixed number of particles is simulated iteratively by following three steps: a) Prediction: Based on the current position of the particle, next position is predicted. b) Measurement: Likelihood of a particle to lie at the current position is estimated. More likely particle is given higher weight c) Re-sampling: Particles with higher weights are replicated and particles with lower weights are eliminated. This method is able to find the actual shape of failure distribution at the MPFP and is also scalable. However, for very small failure probabilities, it converges only after 10^4 simulations, which is still better than the above methods which need more than 10^5 simulations to find the MPFP.

These MPFP+IS based methods try to find MPFP with repeated random sampling even when they are not near the failure boundary, thus wasting simulations. Another limitation is that the reduction in simulations by these methods is not enough when design becomes more complex (i.e. more transistors or more variation sources are added to the design) because the number of simulations increase exponentially with dimensions.

III. Proposed Method: SSFB

The algorithms assumes that the distribution of failure points in the parameter space follows the condition:

If a fail point P_i exists with parameter values $\{P_{ik}$ |k=1::n$\}$, then point P_j with

$$\{P_{jk} \geq P_{ik} | k = 1 :: n\} \; if \; P_{ik} > 0$$
$$\{P_{jk} \leq P_{ik} | k = 1 :: n\} \; if \; P_{ik} < 0 \qquad (2)$$

is also a fail point. For instance in 2-D case, if fail point with parameters values <2σ,2σ> exists, then points <2σ,3σ>,<3σ,2σ> are also fail points. Similarly with fail point <-2σ,-2σ>, point <-2σ,-3σ>,<-3σ,-2σ> are also fail points.

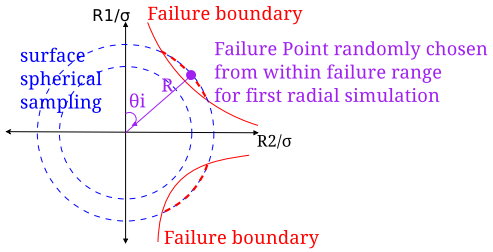

Figure 1. Hyper-Sphere surface sampling to reach a failure point. The figure shows the failure boundary in one hemisphere. The same approach would be applied for the second hemisphere. The Radius of the hypersphere is incremented by 1σ till a failure occurs.

Khalil el. al. [1] described this condition as the monotonic behaviour of SRAM failure in which the bit-cell after failure because of variation, does not return to normal operation with more variation in its parameters. In this paper, we provide the results for read-failure probability. Since the failure boundaries of other SRAM margins also satisfy (2), this proposal is applicable to all SRAM margins.

In the proposed method, we use spherical coordinates for hypersphere to facilitate the computation. The spherical coordinates for n-sphere (which are to be generated uniformly [10]) (i.e. 'n' dimensional hypersphere) consist of

1) Radial coordinate, R

2) $n-1$ angular coordinates $\theta_1, \theta_2 ... \theta_{n-1}$, where $\theta_{n-1} \in [0, 2\pi]$ and all other $\theta_i \in [0, \pi]$

Figures 1-3 give a demonstration of SSFB for the 2-dimensional case. The steps involved in SSFB are described in detail below:

A. Hypersphere surface sampling

Uniform sampling is done on the surface of hypersphere with initial radius R_0 to find failure points. If there is no failure point among the samples then the radius of hypersphere is incremented by 1σ. The objective of this step is to reach a failure point quickly. Since the sampling is done on the surface of hypersphere, all failure points are at the same radial distance from the origin. Hence, each of these failure points has the same failure-probability. We choose a random point from these failure points as our first failure point for radial simulation.

B. Radial simulation

The fail point chosen is simulated radially inwards till the point reaches the pass/fail boundary. It is easy to know when a point has reached the pass/fail boundary as the next radial simulation of that point will give a pass result. This is shown in figure 2, where the radial simulation of point stops once the point passes the failure condition. The last failed location of the point in the radial simulation is then used for spherical surface sampling in the next step.

C. Spherical-Surface Sampling

Once at the pass/fail boundary, there is no point in doing further radial simulations in that direction. Instead the simulation

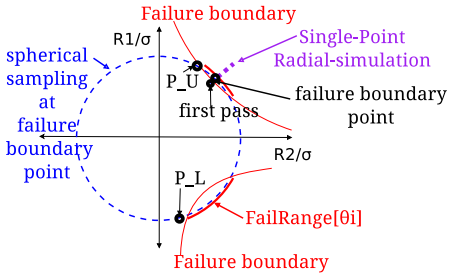

Figure 2. The radial simulation and the subsequent spherical sampling at Pass/Fail boundary point. Fail Ranges are identified in each quadrant of the hemisphere. The upper bound (P_U) and the lower bound (P_L) on the failure range are identified.

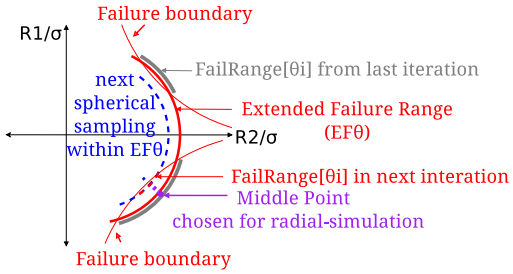

Figure 3. The Extended-Failure Range [EF] and next spherical sampling. The EF range, defined as [P_U,P_L] covers fail ranges in both quadrants and also, some part of the non-failure region of the hypersphere surface. For next iteration, middle point from the largest FailRange is radially simulated till it reaches pass/fail boundary. The next spherical sampling is done within the last iteration's EF range.

should be shifted to a different point at the same radius that has the maximum probability of being close to MPFP. A surface-spherical random sampling is thus performed for a hypersphere with radius as the norm of the current failure boundary point and center as its origin. The objective of this random sampling is not to choose the minimum-norm point from the failed sampled points. Rather, it is to find an approximate range of failure points in each angular dimension at the current radius. Then, from the largest fail range the middle failed point is chosen as an approximation to the closest point to MPFP. This is shown in figure 3. Initially, it is not required to find the failure-ranges with high accuracy and so random sampling can be done for smaller number of samples. As we move closer and closer to the origin, the failure-range decreases, thereby increasing the simulation density. So, the accuracy of range estimation increases as the simulated point moves closer to the origin.

The failed sampled points from the spherical sampling are classified based on their hemisphere i.e. for a failed point whether its $\theta_{n-1} \in [0, \pi]$ or $[\pi, 2\pi]$. For a dimension θ_i, the fail points are sorted by θ_i and stored as FailRange[θ_i]. Then the maximum failure-range for θ_i, $MaxFail[\theta_i]$ is defined as the $Max(FailRange[\theta_i]_{hemisphere1}, FailRange[\theta_i]_{hemisphere2})$. $MaxFail[\theta_i]$ consists of two fail point lists for each quadrant in $[0, \pi]$.

Some issues arise while finding the failure-range as dimensions of the problem increase:

1) If the failure-range in each quadrant is found independently then we have to consider failure-ranges that increase exponentially in number with increasing dimensions. To solve this dimensionality problem, we define for a dimension θ_i:

$$ExtendedFailureRange,\ EF[\theta_i] = [P_U, P_L] \qquad (3)$$

where, P_U : the last pass before the first fail in $MaxFail[\theta_i]$, is the upper bound on failure-range and,

P_L: the first pass before the last fail in $MaxFail[\theta_i]$, is the lower bound on the failure-range

This $EF[\theta_i]$ range is inclusive of failure-ranges in all quadrants for a given dimension θ_i, with some non-failure points as well. The $EF[\theta_i]$ range for our demonstration example is shown in figure 3 . This trade-off allows for linear scaling of failure-ranges as the number of dimensions increase because of each dimension we only have one range $EF[\theta_i]$.

2) Finding the next point for radial simulation becomes a challenge with $EF[\theta_i]$ because it also includes non-failure points and so the middle point from the range cannot be chosen as the next point for radial simulation. To solve this issue, we only save the failure-range $MaxFail[\theta_i]$ of the dimension θ_i with largest $EF[\theta_i]$. Starting at this failure-range, we choose the middle point for radial simulation in subsequent iterations: Random uniform surface-sampling in the next iteration for each dimension θ_i is done only within its extended failure-range $EF[\theta_i]$. This is based on the observation that the extended failure-range $EF[\theta_i]_{iteration=j} \subseteq EF[\theta_i]_{iteration=j-1}$ which can be seen in figure 3.

D. Termination

When the radially simulated point reaches the pass/fail boundary and the subsequent spherical sampling results in no failures, then it is an indication that the point is close to the MPFP. At this stage, we decrease the step width for radial simulation by one-eight and continue with radial simulation till the point crosses the pass/fail boundary. Random sampling is performed again at this new found fail boundary point and if none of the sampled points fail then the step-width is again decreased by one-eight. This continues till the step-width becomes smaller than 0.01σ.

The reason for dividing by a large factor (i.e. 8) is based on the observation that if the last failure boundary point found is the MPFP, then the random spherical sampling is needed only three times with an overhead of 8 simulations before each sampling. While if the step-width were to be half-ed, then the random spherical sampling would have to be done 8 times resulting in more overall simulations.

IV. Application to SRAM Analysis

The variability in SRAM is modeled as a threshold voltage variation. Threshold voltages are modeled as normal distributions with mean as the nominal model value and sigma as 10% of nominal value. The simulations are done for 32nm PTM

Algorithm 1 SSFB

$StepSize := 1$
$R := R_0$ # Radius of Hypersphere
$EF_\theta := [(0,\pi)_{\theta_1},\ (0,\pi)_{\theta_2}...(0,\pi)_{\theta_{n-2}},\ (0,2\pi)_{\theta_{n-1}}]$
while no failure found **do**
  $R \leftarrow R + 1$
  Uniform Surface Sampling on EF_θ
end while
P :=Random Chosen Fail Point
while $StepSize \not< 0.01$ **do**
  while $P \neq Pass$ **do**
    Decrement Radius(P) by $StepSize$
  end while
  $R := Radius(P)$
  Uniform Surface Sampling on EF_θ
  Find Fail Points
  Separate Fail points for two Hemispheres
  (θ_{n-1}:[0,π],[π,2π])
  Foreach θ_i **do**
    Fail[θ_i] :=Sorted Fail points on θ_i
      classified by their Hemisphere
      and Quadrant
    MaxFail[θ_i]=Max(Fail[$\theta_i]_{HEM1}$,Fail[$\theta_i]_{HEM2}$)
    P_u :=Last Pass before First fail in MaxFail[θ_i]
    P_L :=First Pass after Last fail in MaxFail[θ_i]
  end Foreach
  $\theta_{Max} := \theta_i$ with Max ($[P_U, P_L]$)
  $EF_\theta := [(P_U, P_L)_{\theta_1}...(P_U, P_L)_{\theta_{n-1}}]$
  P :=Middle point in Fail[θ_{MAX}]
  IF $EF_\theta \equiv None$ **do**
    $StepSize := StepSize/8$
    P :=Last Fail Point
  end IF
$MPFP := P$

process in HSPICE circuit simulator. SSFB is compared with IHS-DHS and Mean-Shift methods. Each method is executed 20 times to compute the variance of the results.

A. SRAM analysis setup

The static noise margins analysis of the bit cell are computed through the N-Curve method proposed in [2] . The read noise margins provided by N-curve are:

SVNM Read Static-Voltage-Noise-Margin is the maximum amount of DC voltage noise from the bit-lines that the storage node storing value '0' can handle before flipping to value '1'.

SINM Read Static-Current-Noise-Margin is the peak current noise from the bitlines that the storage node storing value '0'

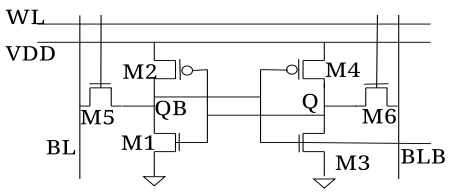

Figure 4. The schematic of 6T-SRAM bitcell. M1 and M3 are pull-down transistors. M2 and M4 are pull-up transistors. And M5 and M6 are access transistors driven by wordline WL and connected to bitlines BL and $\bar{BL}$.

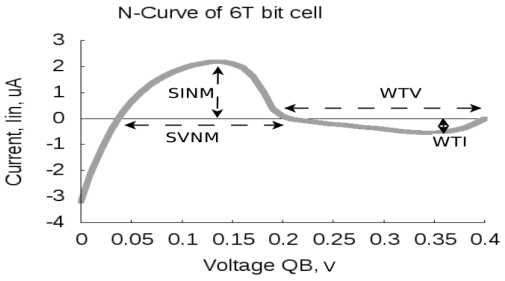

Figure 5. N-curve for 6T bitcell showing the definitions of noise margins. The read margins are SINM and SVNM. The write margins are WTI and WTV.

can handle before the voltage at that node rises to above the threshold of the inverter.

The schematic of 6T bitcell is shown in figure 4 along with its N-curve in figure 5. The read-failure probability for this analysis is defined as:

$$P_{ReadFail} = Prob(SVNM < V_T || SINM < 0) \quad (4)$$

where, V_T is the thermal voltage (26mV at 300K), which is used as lower bound because for $SVNM < V_T$, thermal noise can cause bitcell to flip.

B. Number of Samples

To find the minimum number of samples that are sufficient during spherical surface sampling, we compared the results of using 100 to 1000 samples per random sampling. The radial distance of the MPFP and the number of simulations averaged over 20 repetitions are compared in figure 6.

Figure 6 shows that increasing the number of samples above 100 does not give a large increase in accuracy($< 0.1\sigma$). The focus of this paper is on decreasing the simulation time; hence, we use only 100 samples per random sampling for read-failure analysis.

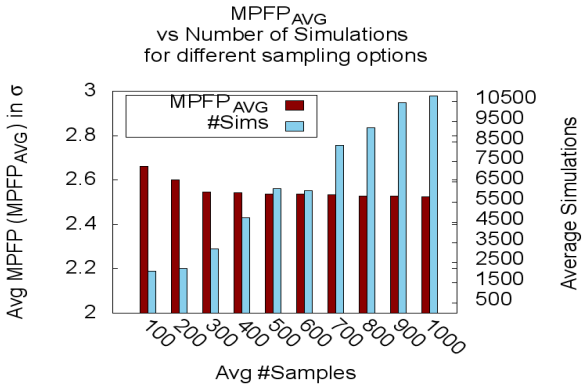

Figure 6. Mean Radial Distance of MPFP vs Total simulations for different sampling options. The radial distance is normalized to σ

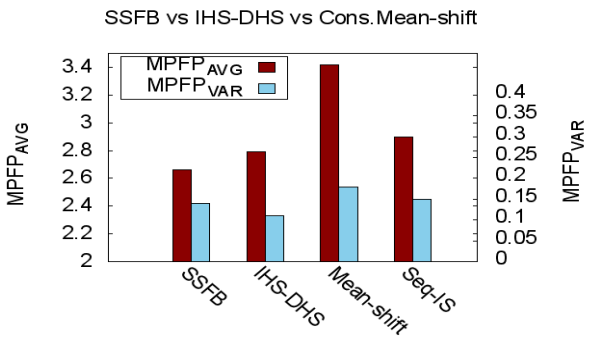

Figure 7. Accuracy Comparison of Proposal with IHS-DHS and mean-shift method. MPFP_{AVG} is the mean radial distance of MPFP and MPFP_{VAR} is the variance of radial distance of MPFP for 20 different runs.

Table I. AVERAGE NUMBER OF SIMULATIONS FOR FINDING MPFP

MPFP estimation Method	Average #Simulations	Runtime
Proposal	2078	1m33s
IHS-DHS	8.3e+4	7m37s
Mean-Shift	7.7e+4	6m11s
Seq-IS	4728	3m57s
Runtime on 4-thread 2-core 2.5GHz processor, L1-32KB, L2-256KB		

C. Read-Failure Probability

The read-failure probability as defined in eq.4 is calculated using the SSFB method and is compared with the methods IHS-DHS, Mean-Shift and Seq-IS. We simulate a 6T bitcell operating at 0.6V. The read-failure analysis is repeated 20 times to compare the variance of MPFP estimate for these methods. For IHS-DHS simulations, 10^4 samples are simulated for each IHS and DHS. For Mean-Shift method, $2.5e+4$ simulations are performed during the first hypersphere sampling and for rest of the steps 10^4 simulations are performed. For Seq-IS method, sampling-resampling algorithm is run with 500 particles.

The results in figure 7, show that the proposed method has a similar variance as the Mean-Shift method and a smaller variance than the Seq-IS method. Also, the proposed method needs 40x less simulations (table 1), thus it saves computing power and time. Since, SSFB always terminates exactly at the pass/fail boundary, the estimate of the MPFP point by the SSFB is the lowest of all. Another non-MPFP and non-IS based such as the machine learning based classifier approach [5] needs more than 40,000 simulations for building the classifier and in its learning phase. Thus SSFB approach has the smallest simulation time of any of the known approaches to date.

V. CONCLUSIONS

In this paper, we proposed SSFB: a method that can efficiently search MPFP by reducing the sampling range iteratively and sampling only on the surface of hypersphere. To compare the effectiveness of SSFB with other methods, its SRAM read-failure analysis results were compared with IHS-DHS and Mean-Shift methods. SSFB gave smaller norm MPFP than the other methods with a reduction of 40x in the number of simulations. Smaller norm MPFP gives better results with Importance-Sampling as the alternate distribution has a smaller difference to the original distribution. Another advantage of the SSFB method is the better scalability of the number of simulations with increasing dimensions. This is due to the sampling method chosen where the points are sampled uniformly from the surface of hypersphere rather than from within the hypersphere.

VI. ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Education and Science under grant TIN2010-18368, Generalitat of Catalunya under grant 2009SGR1250 and Intel Corporation. Manish Rana is supported by FI-DGR-2013-638 grant.

REFERENCES

1) D. Khalil, M. Khellah, N. Kim, Y. Ismail, T. Karnik, V. K. De, and S. S. Member, "Accurate Estimation of SRAM Dynamic Stability," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16, no. 12, pp. 1639–1647, Dec. 2008.
2) E. Grossar, M. Stucchi, and K. Maex, "Read stability and write-ability analysis of SRAM cells for nanometer technologies," Solid-State Circuits,, vol. 41, no. 11, pp. 2577–2588, 2006.
3) K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi, and T. Sato, "Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis," pp. 703–708, Nov. 2010.
4) L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization," 2008 IEEE/ACM International Conference on Computer-Aided Design, pp. 322–329, Nov. 2008.
5) L. Sun, J. Mathew, D. K. Pradhan, and S. P. Mohanty, "Statistical Blockade Method for Fast Robustness Estimation and Compensation of Nano-CMOS Arithmetic Circuits," 2011 International Symposium on Electronic System Design, pp. 194–199, Dec. 2011.
6) M. Qazi, M. Tikekar, L. Dolecek, D. Shah, and A. Chandrakasan, "Loop flattening & spherical sampling: Highly efficient model reduction techniques for SRAM yield analysis," 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), pp. 801–806, Mar. 2010.
7) R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events," in 2006 43rd ACM/IEEE Design Automation Conference, 2006, pp. 69–72.
8) T. Date, S. Hagiwara, K. Masu, and T. Sato, "Robust importance sampling for efficient SRAM yield analysis," 2010 11th International Symposium on Quality Electronic Design (ISQED), pp. 15–21, Mar. 2010.
9) T. Kida, Y. Tsukamoto, Y. Kihara, R. E. Corporation, and Y. K. Renesas, "Optimization of importance sampling Monte Carlo using consecutive mean-shift method and its application to SRAM dynamic stability analysis," in Thirteenth International Symposium on Quality Electronic Design (ISQED), 2012, no. x, pp. 572–579.
10) http://mathworld.wolfram.com/SpherePointPicking.html