# Leveraging Sensitivity Analysis for Fast, Accurate Estimation of SRAM Dynamic Write $V_{MIN}$

James Boley[†], Vikas Chandra[‡], Robert Aitken[‡], Benton Calhoun[†]
[†]Dept. of ECE, University of Virginia, Charlottesville; [‡]ARM R&D, San Jose
{jmb9zw,bcalhoun}@virginia.edu {vikas.chandra,raitken}@arm.com

**Abstract—Circuit reliability in the presence of variability is a major concern for SRAM designers. With the size of memory ever increasing, Monte Carlo simulations have become too time consuming for margining and yield evaluation. In addition, dynamic write-ability metrics have an advantage over static metrics because they take into account timing constraints. However, these metrics are much more expensive in terms of runtime. Statistical blockade is one method that reduces the number of simulations by filtering out non-tail samples, however the total number of simulations required still remains relatively large. In this paper, we present a method that uses sensitivity analysis to provide a total speedup of ~112X compared with recursive statistical blockade with only a 3% average loss in accuracy. In addition, we show how this method can be used to calculate dynamic $V_{MIN}$ and to evaluate several write assist methods.**

## I. INTRODUCTION

Static Random Access Memory (SRAM) is a critical component of today's SoCs consuming large amounts of area and often setting the critical timing path. Technology scaling has allowed reductions in area, power, and delay. In order to continue this trend, the minimum operating voltage ($V_{MIN}$) of SRAMs must continue to scale down. This has become increasingly difficult as devices enter the nanoscale range due to increased device variability and leakage. SRAM devices are typically minimum sized, which further compounds this problem [1]. The increase in both variation and leakage leads to reduced read and write margins, making it more difficult to design low power SRAMs that meet frequency and yield constraints. In addition, as the capacity of SRAM arrays continues to increase, the stability of the worst case bitcell degrades. Therefore it has become increasingly important to accurately predict SRAM yield at a given supply voltage.

The most common method for evaluating yield is through Monte Carlo (MC) simulations. However for very large arrays (i.e. 10 Mb) the number of simulations required to identify the worst case bitcell becomes prohibitively large. Because the majority of simulated samples do no lie in the tail region, a full MC simulation is not an efficient method for estimating very small failure probabilities. A common approach to reducing simulation time is to run a relatively small number of samples and then fit the resulting distribution to the normal distribution. Once the $\mu$ and $\sigma$ are known, the stability of the worst case bitcell can be identified. The problem with this approach is that it can only be applied to data sets that replicate a known distribution [2][3]. However, it has been recognized that the dynamic write margin does not fit the normal distribution [3][6]. The distribution resembles the long tail F-distribution, but does not match it exactly. Because the distribution does not closely match any known statistical distribution, it is difficult to model without full simulation of the tail region.

One approach to solve this problem is to develop purely analytical models as in [4][5]. However these approaches are less accurate because approximations must be made to simplify the problem. [6] showed that these approximations can lead to errors in failure probability estimates of up to three orders of magnitude. Two methods that reduce MC run time by effectively simulating only points in the tail region include importance sampling [7][8] and statistical blockade [9][10]. These techniques can be used to reduce simulation time by several orders of magnitude. However, because the calculation of the dynamic margin requires a much larger number of simulations than the static noise margin (SNM), these methods still require long simulation times. In this paper we present a methodology using sensitivity analysis to further reduce the time required to calculate dynamic write $V_{MIN}$.

The rest of this paper is organized as follows. Section II provides background information on the dynamic metric used to quantify write-ability. In section III we present the methodology. Section IV shows how this methodology can be used to calculate dynamic write $V_{MIN}$. Section V applies the methodology to evaluate write assist methods and section VI concludes. We use a sub-32nm commercial CMOS bulk process for all of the simulations in the paper.

## II. BACKGROUND

The dynamic noise margin is defined as the minimum pulse width required to write the cell, or $T_{CRIT}$ [11-16]. The benefit of this metric is that it takes into account the transient behavior of the bitcell, which is not captured by static metrics. This metric has been shown by [14] to produce more accurate $V_{MIN}$ estimations than static metrics, since static metrics give optimistic write margins and pessimistic read margins, due to the infinite wordline (WL) pulse width. In this paper we focus primarily on dynamic write-ability since the static metric results in optimistic yields and because it has been shown that write failure is more likely in newer technologies [17]. The downside to using transient simulations is that they are more time costly, especially when running large numbers of Monte Carlo samples to isolate the worst case bitcells. Whereas

978-3-9815370-0-0/DATE13/©2013 EDAA

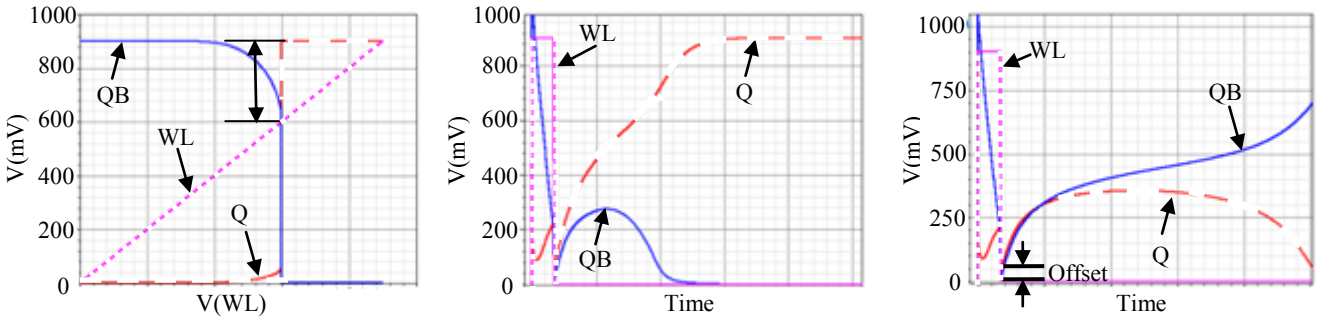


Figure 1. a) DC sweep of WL allows for the write margin to be calculated in a single simulation, b) successful write operation c) even with QB pulling below Q at the end of the WL pulse, the write is not successful

the static margin can be calculated using a single simulation (Figure 1a), the calculation of $T_{CRIT}$ requires a binary search. This takes on average ten to fifteen iterations to accurately determine the critical pulse width with a high level of precision. Figure 1 b-c shows that in the presence of variation, pulling QB below Q doesn't guarantee a successful write.

In [2][3] the author defines static $V_{MIN}$ under the presence of variation. $V_{MIN}$ is defined as the point where the SNM becomes zero. The author uses the hold SNM to define the data retention voltage, the read SNM to define read $V_{MIN}$, and the WL sweep method to define write $V_{MIN}$ [18]. To estimate the failure probability at a given supply voltage, each metric is simulated across a range of $V_{DD}$s. Each resulting distribution is then fitted to the normal distribution. As $V_{DD}$ is reduced, the mean of the write distribution decreases and the standard deviation increases. Then using equations (1) and (2), the failure probability can be calculated for any $V_{DD}$. In equation (1), $s$ is equal to the SNM which causes a failure, which in this case is just zero. $\mu_l$ and $\mu_h$ are defined as the SNM for writing a zero and writing a one. Equation (2) is a best fit line representing the value of $\mu$ and $\sigma$ versus $V_{DD}$.

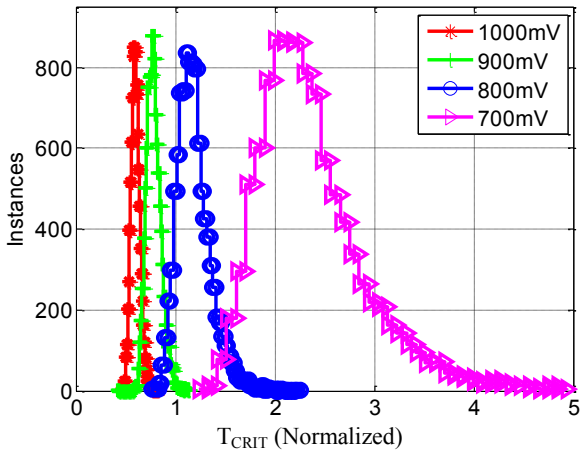


Figure 2. The distribution of $T_{CRIT}$ does not fit a normal distribution

$$p_f = \frac{1}{2}erfc\left(\frac{\mu_{h-s}}{\sqrt{2}\sigma_0}\right) + \frac{1}{2}erfc\left(\frac{\mu_{l-s}}{\sqrt{2}\sigma_1}\right) - \frac{1}{4}erfc\left(\frac{\mu_{h-s}}{\sqrt{2}\sigma_0}\right) * \frac{1}{4}erfc\left(\frac{\mu_{l-s}}{\sqrt{2}\sigma_1}\right) \quad (1)$$

$$\mu = \mu_0 + a(v^2 - v_0^2) + b(v - v_0), \quad \sigma = \sigma_0 + c(v - v_0) \quad (2)$$

The problem with this approach is that the dynamic margin is not normally distributed. From Figure 2, the shape of the $T_{CRIT}$ distribution is long tailed, making the normal approximation inaccurate. In order to apply a similar method as in [2][3], a new distribution must be identified that fits the data. Using the curve fitting toolbox in MATLAB, we were able to determine that the $T_{CRIT}$ distribution closely matches the Frechet distribution, whose probability density function is shown in (3), for $V_{DD} \leq 800$ mV.

$$f(x) = \frac{\alpha}{\beta}\left(\frac{\beta}{x-\mu}\right)^{\alpha+1} \exp\left[-\left(\frac{\beta}{x-\mu}\right)^{\alpha}\right] \quad (3)$$

The Frechet distribution is an extreme value distribution, commonly used to estimate the maxima of long sequences of random variables. The three parameters: $\mu$, $\alpha$ and $\beta$ represent the offset, shape, and scale respectively. When applying the fit to a sample size of 5K Monte Carlo points, we were able to closely match the Monte Carlo data. However, due to the shape of the distribution, the fit software calculates a large confidence interval for the three fitting parameters, resulting in errors modeling the tail region. Figure 3 shows three possible fit lines predicted by the curve fitting tool. The actual line represents data taken using importance sampling to estimate the extremely low failure probabilities. At a certain point, the probability of failure remains constant as the WL pulse width is increased. This is due to the fact that at 500 mV, the memory is approaching the static failure point, and therefore a wider WL pulse does not reduce the failure probability. This figure shows that it is not possible to extrapolate the tail of the distribution using only a small (5K) Monte Carlo sample.

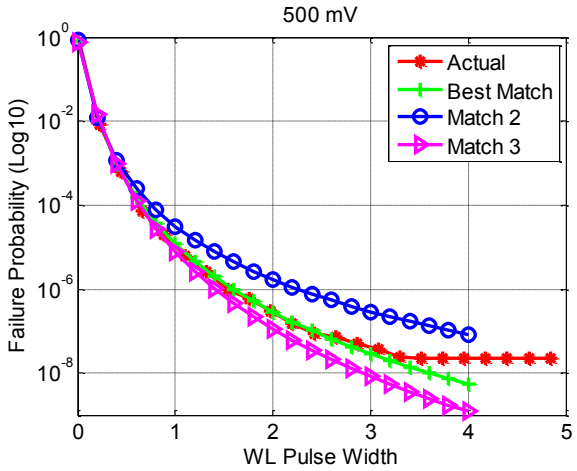


Figure 3. The three distributions match the MC data, however they do not match the tail of the distribution

Another method to determine the dynamic write margin of the worst case bitcell is recursive statistical blockade [10]. However, in order to accurately determine the dynamic margin using binary search, its takes an average of twelve simulations. Using this method, it would take over 894,000 simulations to identify the worst case write margin for a 100 Mb memory. In the next section we will describe a method using sensitivity analysis to accurately predict the worst case bitcell that requires less than 1,100 simulations.

## III. ESTIMATING DYNAMIC WRITE MARGIN ($T_{CRIT}$)

In order to reduce the cost of running large numbers of transient Monte Carlo simulations, we propose using sensitivity analysis to quickly generate the $T_{CRIT}$ distribution [19]. The first step in this method is to sweep the threshold voltages of each transistor to produce the plot shown in Figure 4. The PU, PD, and PG labels represent

$$T_{\text{CRIT-OFFSET}} = aV_{T-Shift}^3 + bV_{T-Shift}^2 + cV_{T-Shift} + d \qquad (4)$$

the pull-up, pull-down, and passgate transistors respectively. The left node of the bitcell is initially holding a '0' and the right node is initially holding a '1'. The x-axis represents the $V_T$ shift of each transistor ranging from -6σ to 6σ; the y-axis represents the resulting $T_{CRIT}$ value. When sweeping the $V_T$ of each transistor, all other transistors are left at nominal $V_T$. We then fit each curve to a third order polynomial. Once each of the curves has been fitted, the next step is to generate a $V_T$ distribution for each of the six transistors (Figure 5). This is done by generating a normal distribution using the sigma values from the Spice model. Next, the $V_T$ offset of each transistor is plugged into (4), and the six offsets are then added to the nominal case to produce the $T_{CRIT}$ prediction:

$$T_{\text{CRIT}} = T_{\text{CRIT-NOM}} + T_{\text{CRIT-OFFSET-PUL}} + \ldots + T_{\text{CRIT-OFFSET-PGR}} \qquad (5)$$

This calculation is repeated N times depending on the desired sample size. Clearly, computing (5) is much faster

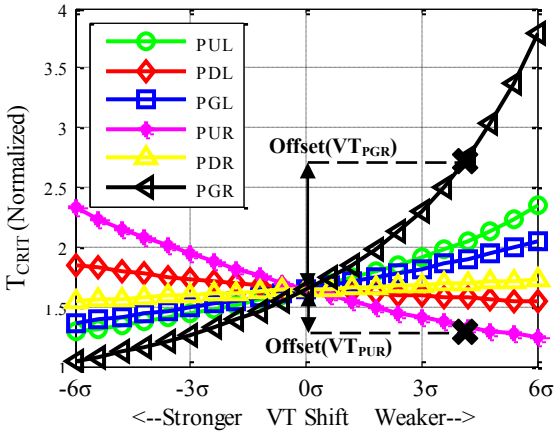


Figure 4. In order to characterize the bitcell, the VT of each transistor is swept independently

than running the set of simulations required to find $T_{CRIT}$ using Spice.

One assumption made by sensitivity analysis is that the $V_T$ variation of each transistor has an independent effect on $T_{CRIT}$. In order to verify this, we repeat the $V_T$ sweep on each transistor, adding Monte Carlo variation to the other five transistors. If the $V_T$ variation of each transistor does have an independent effect on $T_{CRIT}$, then we would expect the shape of the $V_T$ curve to remain the same in the presence of variation. As shown in Figure 6, the shape of the $V_T$ shift vs. $T_{CRIT}$ curves does not change significantly; the curves are simply shifted up or down from the nominal case. This is true for all six transistors. There is some slight overlap between the curves which leads to small errors in the predicted value.

In order to verify the accuracy of this methodology, we compared the margin of the worst case bitcell calculated by the model to the worst case margin calculated by the

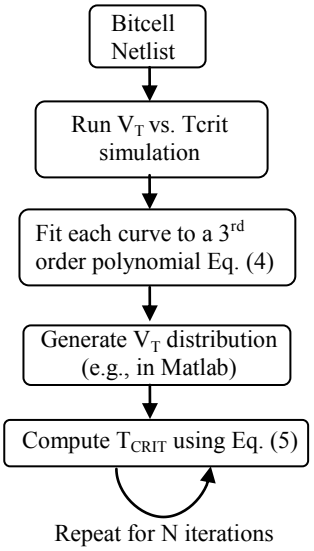


Figure 5. Flowchart of the proposed $T_{CRIT}$ estimation model

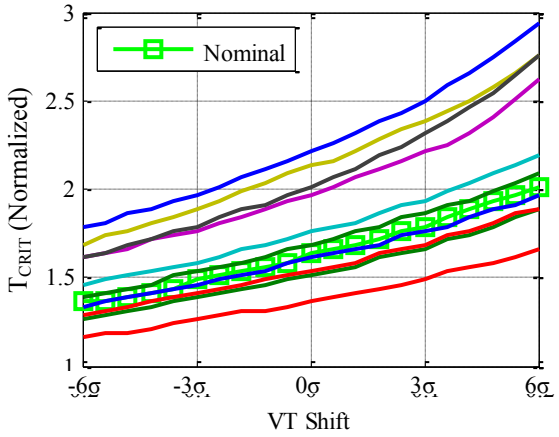


Figure 6. Transistor variation has a close to independent effect on $T_{CRIT}$. Each line represents a single Monte Carlo iteration

recursive statistical blockade tool [10]. The accuracy of the model was tested for three memory sizes: 100 Kb, 10 Mb, and 100 Mb. The model was also tested across a range of $V_{DD}$s from 500 mV up to 1V. The results are shown in Table 1. We can see from the table that the worst case error is only 6.83%, while the average is 3.01%. A positive percentage error means that the model overestimated the $T_{CRIT}$ value, resulting in slightly pessimistic margins.

The advantage of this method is that it greatly reduces simulation times while sacrificing very little accuracy compared to statistical blockade. This same technique can be applied to importance sampling to reduce the total run time. Simulating the $V_T$ curves in Figure 4 requires approximately 18.8 minutes. Once these curves have been produced, random samples are generated (e.g., by MATLAB) and applied to (5). The run time for the sensitivity analysis increases linearly with the number of samples. The total run time for a 100 Mb memory is only 32 minutes. One disadvantage of the statistical blockade tool is that in order to determine the worst case write margin, two separate test cases must be run: writing a '0' and writing a '1'. This means that two separate filters must

Table 1. Percentage error of the sensitivity analysis versus statistical blockade for varying memory sizes and $V_{DD}$

| Modeled data vs. Statistical Blockade (Percentage Error) | | | |
|---|---|---|---|
| | 100K | 10M | 100M |
| 500 mV | 6.83 | -4.25 | 6.51 |
| 600 mV | 2.96 | -3.69 | 5.61 |
| 700 mV | -0.18 | -2.64 | 4.75 |
| 800 mV | 0.83 | -0.7 | 1.21 |
| 900 mV | -4.5 | 0.83 | 1.43 |
| 1000 mV | -2.72 | -2.2 | -2.27 |
| Average | 3.01 | 2.39 | 3.63 |

Table 2. A comparison of the run times between statistical blockade and sensitivity analysis.

| | Statistical Blockade | Sensitivity Analysis |
|---|---|---|
| | Num. simulations | Run Time |
| Initial Simulation | 24,000 | 18.8 min |
| 100 Kb | 107,904 | 0.72 s |
| 10M | 531,096 | 72 s |
| 100M | 231,288 | 12 min |
| Total Simulations | 894,288 | |
| Total Run Time | 60 Hours | 32 minutes |

be generated, as well as two separate sets of Monte Carlo simulations. The total number of simulations required for the recursive statistical blockade tool is 894,288, corresponding to a total CPU runtime of 60 hours.

In summary, our method provides a 112.5X speedup at the cost of an average loss in accuracy of 3.01% and a worst case loss of 6.83%.

## IV. DYNAMIC WRITE $V_{MIN}$ PREDICTION

Write $V_{MIN}$ is defined as the minimum operating voltage in which the write operation will succeed. We can define this minimum operating point using either static or dynamic metrics. Static write $V_{MIN}$ is defined as the voltage that results in an SNM of zero, meaning that even if the WL is pulsed high for an infinite time period, the write operation will fail. Dynamic write $V_{MIN}$ is defined in [14] as the voltage in which the worst case $T_{CRIT}$ value is larger than the word line pulse width. In order to calculate dynamic write $V_{MIN}$ using our approach, we can repeat the procedure described in Figure 5 for varying $V_{DD}$. In this example we chose six test points between 0.5V and 1V. The procedure can be repeated for different memory sizes,

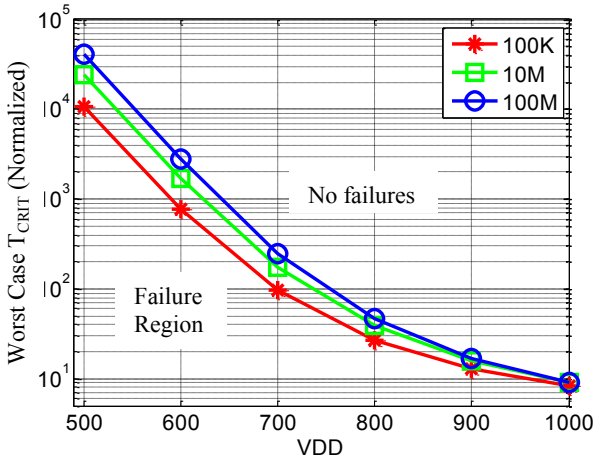


Figure 7. The lines represent the point of single failure while the region above represents no fail, and the region below represents multiple bit fails

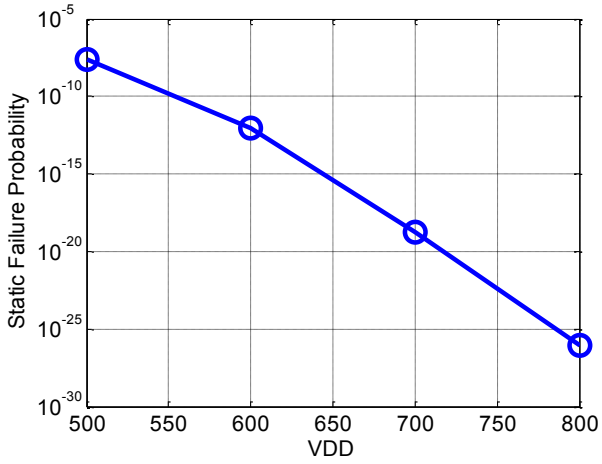


Figure 8. Static failure probability versus $V_{DD}$

and the worst case dynamic write margin can be plotted versus $V_{DD}$.

In the plot on Figure 7, the individual lines represent different memory sizes in bits. The curves represent the ($T_{CRIT}$, $V_{DD}$) point of the first single bit failure. Below the curve represents multiple bit failures and the region above the curve has no bit failures. By choosing a WL pulse width constraint and memory size, we are able to determine the minimum operating voltage necessary to ensure reliable operation. We can see that as the size of the memory increases, the critical word line pulse width for the worst case bitcell also increases. This effect becomes more pronounced as $V_{DD}$ is scaled down. Generating these same results across $V_{DD}$ using statistical blockade would take approximately 360 hours.

To show the importance of using dynamic write $V_{MIN}$ as opposed to static, we have plotted the static failure probability versus $V_{DD}$ in Figure 8. At 500 mV, the static failure probability is 2.57e-8, which means that in a 100 Mb SRAM cache, there will likely be two to three bitcells statically failing. At 600 mV, this failure probability decreases by over five orders of magnitude, meaning that at this operating voltage, it is unlikely that there will be any bitcells failing statically. As $V_{DD}$ is raised, the static failure probability continues to decrease. Clearly margining based on static failure probabilities leads to drastic overestimation of SRAM yield. While dynamic metrics allow for a much more accurate measure of $V_{MIN}$ they are much more costly to simulate than static metrics. This is why having a method to accurately predict dynamic write $V_{MIN}$ is so valuable.

## V. IMPACT OF ASSISTS ON DYNAMIC WRITE $V_{MIN}$

As SRAMs continue to scale, peripheral assist methods will be needed to allow for continued voltage scaling [20-25]. Therefore it is important to determine which write assist methods provide the largest reductions in dynamic write $V_{MIN}$. Some assist features such as cell VSS (CVSS) boost aim to weaken the cross coupled inverters, by

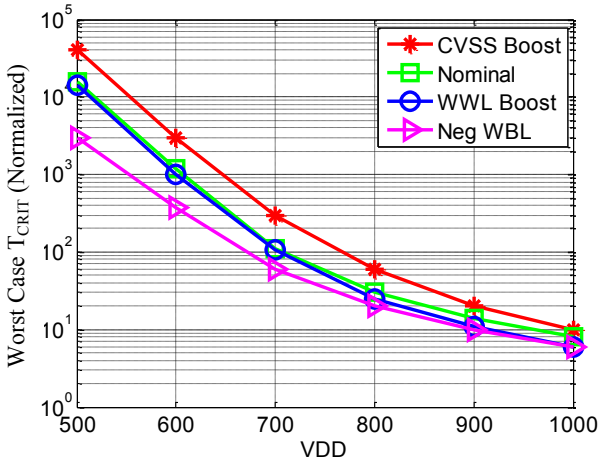


Figure 9. Measuring the effects of write assist methods on dynamic write $V_{MIN}$

reducing the gate to source voltage of the PMOS device. However, we will show that this technique actually increases the worst case $T_{CRIT}$ value. With write WL (WWL) boosting, we increase the gate voltage of the passgate transistor in order to improve the drive strength. The negative BL reduction technique also increases the drive strength of the passgate by dropping the voltage on the source. Using sensitivity analysis, we can quickly and accurately make predictions about which assist methods are the most effective across a range of $V_{DD}$s.

In Figure 9, the memory size is 1 Mb, and the ΔV for each assist method is 100 mV. We can see that the CVSS boost technique actually increases the worst case $T_{CRIT}$ value. This is due to the fact that the weakening of the cross coupled pair results in a longer time delay for the node initially holding a '0' to pull high. At high $V_{DD}$, the negative BL reduction and WWL boost have comparable effects on reducing $T_{CRIT}$, however as $V_{DD}$ is reduced, the negative BL technique provides much larger reductions in $T_{CRIT}$. This can be explained by the $V_T$ curves in Figure 4. We can see from this plot that as the PUR transistor

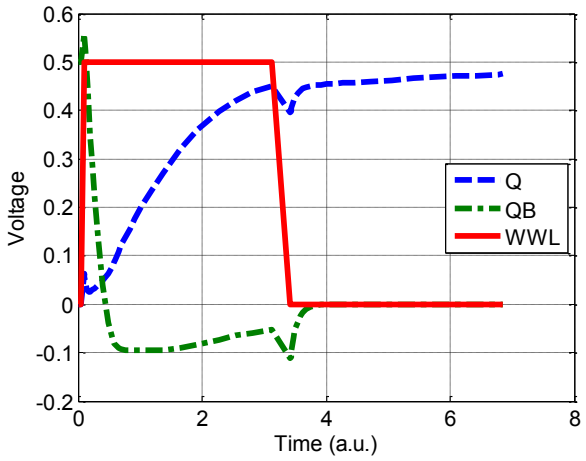


Figure 10. The negative BL reduction results in improved write times due to the QB node being pulled negative

(initially on) gets stronger, the $T_{CRIT}$ value increases as expected. However, as the PUL transistor (initially off) gets stronger, the $T_{CRIT}$ value decreases. This second order effect is due to the fact that as the PUL transistor gets stronger, it is able to more quickly pull the internal node high, resulting in a slightly faster switching time. Because the negative BL technique is passing a stronger '0' (e.g. a negative voltage) into the bitcell, it is strengthening the PUL transistor by increasing its $V_{GS}$. Therefore the effect of negative BL is twofold: it strengthens the passgate transistor as well as the PUL transistor. This second effect is not seen with the WL boost technique because it is not passing a strong '0'. Figure 10 shows that at lower $V_{DD}$ (e.g. 500 mV), the QB node pulls low relatively quickly, while the majority of the write operation is spent waiting for the Q node to pull high. This explains why boosting the WWL has negligible effects on reducing $T_{CRIT}$ as compared to negative BL at lower $V_{DD}$. These results were obtained using the analysis described in section III, resulting in a total speedup of 672X over statistical blockade.

## VI. CONCLUSIONS

We have shown that modeling the tail of the dynamic write margin using a small Monte Carlo simulation is not effective due to the shape of its distribution. While the static noise margin has been shown in [3] to fit the normal distribution, the dynamic write margin follows a skewed long tailed distribution. We found that at $V_{DD}$s below 800 mV, the distribution fits the Frechet distribution, however the tail of the distribution can only be determined by full simulation due to poor confidence in the tail fit.

While statistical blockade is a good method for reducing simulation time, evaluating dynamic $V_{MIN}$ still requires a large number of simulations. We introduced a method using sensitivity analysis that provides a speed up over statistical blockade of 112X with an average percentage error of 3%. This approach allows for rapid assessment of dynamic write $V_{MIN}$ and write assist features. In addition, we determined that negative BL reduction has a greater effect on reducing dynamic $V_{MIN}$ than WL boosting.

## REFERENCES


[1] A. Bhavnagarwala, X. Tang, and J. Meindl "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *JSSC*, pp. 658-665, 2001.

[2] J. Wang, A. Singhee, R. Rutenbar, and B. Calhoun, "Statistical modeling for the minimum standby supply voltage of a full sram array," *ESSCIRC,* pp. 400-403, 2007.

[3] J. Wang and B. Calhoun, "Minimum supply voltage and yield estimation for large srams under parametric variations," *IEEE Transactions of VLSI Systems,* pp. 2120-2125, 2011.

[4] B. Zhang, A. Arapostathis, S. Nassif, and M. Orshansky, "Analytical modeling of sram dynamic stability," *ICCAD*, pp. 315-322, 2006.

[5] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of sram array for yield enhancement in nanosclaed cmos," *IEEE Transactions on Computer –Aided Design of Integrated Circuits and Systems*, pp.1859-1880, 2005.

[6] D. Khalil, M. Khellah, N. Kim, Y. Ismail, T. Karnik, and V. De, "Accurate Estimation of sram dynamic stability," *IEEE Transactions of VLSI Sytems*, pp. 1639-1647, 2008.

[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," *DAC,* pp. 69-72, 2006.

[8] T. Doorn, E. Maten, J. Croon, A. Bucchianico, and O. Wittich, "Importance sampling monte carlo simulations for accurate estimation of sram yield," *ESSCIRC,* pp. 230-233, 2008.

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

[10] A. Singhee, J. Wang, B. Calhoun, and R. Rutenbar, "Recursive statistical blockade: an enhanced technique for rare even simulation with application to sram circuit design," *VLSID,* pp. 131-136, 2008.

[11] M. Sharifkhani and M. Sachdev, "SRAM cell stability: A dynamic perspective," *JSSC,* vol. 44, pp. 609-619, 2009.

[12] W. Dong, L. Peng, and G.M. Huang, "SRAM dynamic stability: theory, variability and analysis," *ICCAD*, pp. 378-385, 2008.

[13] J. Wang, S. Nalam, and B.H. Calhoun "Analyzing static and dynamic write margin for nanometer SRAMs," *ISLPED*, pp. 129-134, 2008.

[14] S. Nalam, V. Chandra, R. Aitken, and B.H. Calhoun, "Dynamic write limited minimum operating voltage for nanoscale SRAMs," *DATE*, pp. 1-6, 2011.

[15] S.O. Toh, Z. Guo, and B. Nikolic, "Dynamic SRAM stability characterization in 45nm CMOS," *IEEE Symposium on VLSI Circuits,* pp. 35-36, 2010.

[16] M. Yamaoka, K. Osada, and T. Kawahara, "A cell-activation-time controlled SRAM for low-voltage operation in DVFS SoCs using dynamic stability analysis," *ESSCIRC,* pp. 286-289, 2008.

[17] A. Bhavnagarwala et al., "Fluctuation limits and scaling opportunites for cmos sram cells," *IEDM*, PP. 659-662, 2005.

[18] Z. Guo, et al., "Large-scale read/write margin measurement in 45 nm CMOS SRAM arrays," *Proc. Symp. VLSI Circuits,* pp.42-43, 2008.

[19] Y. Tsukamoto, et al., "Worst-case analysis to obtain stable read/write dc margin of high density 6t-sram-array with local vth variability," *ICCAD,* pp. 398-405, 2005.

[20] K. Nii, et al., "A 45-nm single-port and dual-port SRAM family with robust read/write stabilizing circuitry under DVFS environment," *IEEE Symposium on VLSI Circuits*, pp. 212-213, 2008.

[21] Y.H. Chen, et al., "A 0.6V 45nm adaptive dual-rail sram compiler circuit design for lower VDD min VLSIs," *IEEE Symposium on VLSI Circuits,* pp. 210-211, 2008.

[22] M. Iijima, K. Seto, M. Numa, A. Tada, T. Ipposhi, "Low power sarm with boost driver generating pulsed word line voltage for sub-1V operation," *Journal of Computers*, pp. 34-40, 2008.

[23] N. Shibata, H. Kiya, S. Kurita, H. Okamoto, M. Tan'no, T.A. Douseki, "A 0.5V 25 MHz 1-mw 256-kb MTCMOS/SOI SRAM for solar-power-operated portable personal digital equipment – sure write operation by using step-down negatively overdrive bitline scheme," *IEEE J. Solid State Circuits,* pp. 728-742 2006.

[24] R.W. Mann, et al., "Impact of circuit assist methods on margin and performance in 6t sram," *Solid State Electronics,* pp. 1398-1407, 2010.

[25] V. Chandra, C. Pietrzyk, and R. Aitken, "On the efficacy of write-assist techniques in low voltage nanoscale srams," *DATE,* pp. 345-350, 2010.