Statistical Analysis and Design of Subthreshold Operation Memories

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Arquitectura de Computadors

Manish Rana

Department of Computer Architecture
Universitat Politecnica De Catalunya, Barcelona

A THESIS SUBMITTED IN THE FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

June, 2016

Statistical Analysis and Design of Subthreshold Operation Memories

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Arquitectura de Computadors

Manish Rana

Advisors:
Ramon Canal Corretger
Universitat Politecnica De Catalunya
Esteve Amat Bertran
Institut de Microelectrònica de Barcelona (CSIC)

Dedicated to my parents.

Abstract

This thesis presents novel methods based on a combination of well-known statistical techniques for faster estimation of memory yield and their application in the design of energy-efficient sub-threshold memories. The emergence of size-constrained Internet-of-Things (IoT) devices and proliferation of the wearable market has brought forward the challenge of achieving the maximum energy efficiency per operation in these battery operated devices. Achieving this sought-after minimum energy operation is possible under sub-threshold operation of the circuit. However, reliable memory operation is currently unattainable at these ultra-low operating voltages because of the memory circuit's vanishing noise margins which shrink further in the presence of random process variations. The statistical methods, presented in this thesis, make the yield optimization of the sub-threshold memories computationally feasible by reducing the SPICE simulation overhead.

We present novel modifications to statistical sampling techniques that reduce the SPICE simulation overhead in estimating memory failure probability. We target the Most-Probable-Failure-Point (MPFP) based mean-shift Importance Sampling technique for its ease of implementation and provide the shift vector for this sampling technique in far fewer SPICE simulations than the existing approaches. Further improvement in reducing the SPICE simulations is obtained with a sequential sampling scheme. In this scheme, an estimate of the failure and non-failure regions is maintained and updated as new samples are simulated. The sampling scheme finds the optimal sampling regions that are most likely to be sampled under the Importance Sampling and provide the largest increase in the failure/ non-failure estimates. This sampling scheme provides 10x reduction in the SPICE simulations compared to the existing proposals.

We then provide a novel method to create surrogate models of the memory margins with better extrapolation capability than the traditional regression methods. These models, based on Gaussian process regression, encode the sensitivity of the memory margins with respect to each individual threshold variation source in a one-dimensional kernel. The predictions are made using an additive kernel which is the sum of these one-dimensional kernels. We find that our proposed additive kernel based models have 32% smaller out-of-sample error (that is, better extrapolation capability outside training set) than using the six-dimensional universal kernel like Radial Basis Function (RBF).

The thesis also explores the topological modifications to the SRAM bitcell to achieve faster read operation at the sub-threshold operating voltages. We present a ten-transistor SRAM bitcell that achieves 2x faster read operation than the existing ten-transistor sub-threshold SRAM bitcells, while ensuring similar noise margins. The SRAM bitcell further provides 70% reduction in dynamic energy at the cost of 42% increase in the leakage energy per read operation.

Finally, we investigate the energy efficiency of the eDRAM gain-cells as an alternative to the SRAM bitcells in the size-constrained IoT devices. First, we study the minimum energy operation of the 2T and 3T1D gain cells under the absence of process variations. We find that reducing their write path leakage current is the only way to reduce the read energy at Minimum Energy operation Point (MEP). Up-sizing the read path transistors to reduce read delay and increase retention time, on contrary, increases read energy at MEP. Further, we study the effect of transistor up-sizing under the presence of threshold voltage variations on the mean MEP read energy by performing statistical analysis based on the ANOVA test of the full-factorial experimental design. We provide 95% confidence intervals for the difference in the mean MEP read energy achieved by the various up-sized gain cell designs.

Acknowledgements

First and Foremost, I would like to first thank my advisors Professor Ramon Canal and Esteve Amat for their guidance and support during my doctoral studies. I owe my gratitude to them for giving me the freedom and time to explore research ideas that interested me.

The discussions with Ramon, were instrumental in helping me focus on a single idea and develop it into a well-defined research problem. His clarity in thought and communication of research ideas, has had the most positive impact on me in these formative years as a research student. I am thankful for the excellent example he has provided as a researcher and mentor. Esteve has always been available to discuss the interesting research problems steering me in the right direction whenever I needed it. I would also like to express my thanks to Professor Antonio Rubio for mentoring me along with Ramon and Esteve during the last two years.

I am grateful to Jaume Abella (Barcelona Supercomputing Center) and Zoran Jaksic, Roser Rius (UPC) for appreciating the thesis material and for their valuable feedback that improved the thesis as a whole.

I am particularly thankful to Professor Jie Han and Professor Bruce Cockburn at University of Alberta for having me as a visiting student in their research group. Many thanks to HiPEAC for funding my stay during this collaboration period. Professor Bruce allowed me to audit his class on Gaussian Process modeling for which I really grateful.

I am also thankful to all the students who have worked at D6-113 over the last five years. They have contributed immensely to my personal time at UPC. I shared a lot of memorable moments with Gaurang and Rakesh, interesting discussions with Shrikanth and coffee breaks with Sudhanshu.

I would also like to thank all the members of ARCO group, specially the director Professor Antonio Gonzalez for accepting me as a doctoral student under the research group.

I gratefully acknowledge the Generalitat de Catalunya for funding me under FI-DGR doctoral grant that made my thesis possible.

I am in debt to my parents for their love and encouragement and for raising me with a love for science. Thank you!

Barcelona, 21st July, 2016

Acronyms

ANOVA	Analysis of Variance
CDF	Cumulative distribution function
CI	Confidence Interval
DVS	Dynamic voltage scaling
e-DRAM	Embedded dynamic read access memory
GP	Gaussian process
GPR	Gaussian process regression
Ioff	Transistor off current (leakage current)
Ion	Transistor on current
IoT	Internet of things
IS	Importance Sampling
LHS	Latin hypercube sampling
MC	Monte Carlo simulation
MEP	Minimum energy point
MPFP	Most probable failure point
ProbFail	Memory failure probability
PTM	Predictive technology model
RBF	Radial basis function kernel
RBL	Read bitline
RT	Retention time
RWL	Read wordline
SNM	Static noise margin
SPICE, HSPICE	Electronic circuit simulator
VDD	Supply voltage
VGS	Gate-Source voltage
Vth	Threshold voltage

Contents

List of Figures

List of Tables

The light at the end of the tunnel may be an incoming train...

Dilbert

1 Introduction

1.1 Prevalence of always-on low-energy battery operated devices

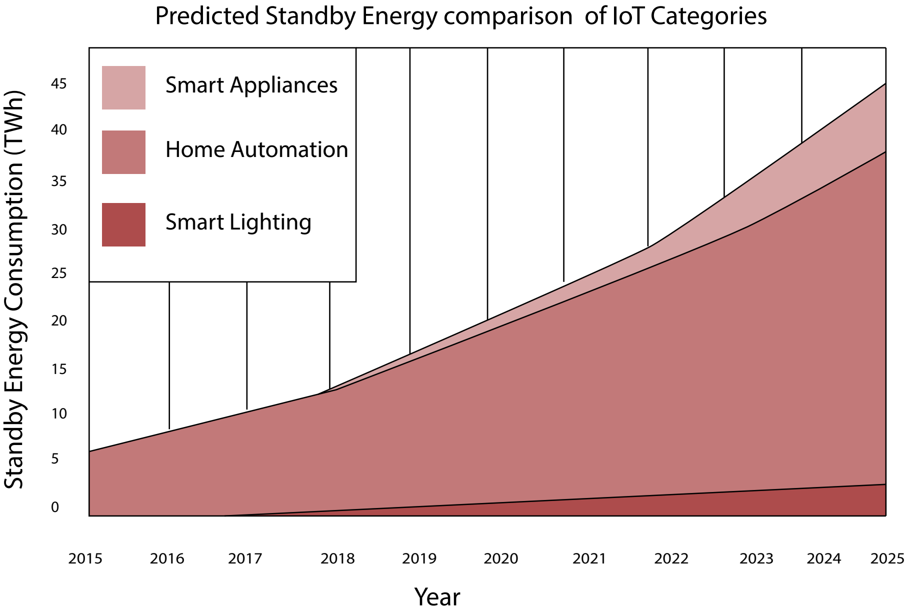


Figure 1: Predicted increase in the standby energy consumption of IOT devices. Source [Friedli, 2016].

The emergence of wireless Internet-of-Thing devices (IoT) has made the "anywhere anytime" computing paradigm a reality. McKinsey Global Institute report [Manyika et al., 2011] predicts that 50 billion battery operated IoT devices will be connected in the cloud by 2020, providing facilities such as smart security, smart health, smart home and many more in domains which were till yet silicon free. The IHS assessment report on wearable technology market [Walker, 2013] estimated roughly 120 million units sold in 2013 generating a revenue of $10 billion. The wearable market is predicted to generate a revenue of over $32 billion in the year 2019.

However, the always-on requirement on these devices raises the concern about the energy consumption of these devices during operation and idle states. Their smaller form factor size and tighter restriction on the weight have led to the use of small capacity batteries. It is in interest to have longer battery life for these devices that forgoes frequent daily/weekly re-charging of its battery. It should also be noted that per person these devices are predicted to increase, all of them always connected to the cloud. As such energy budget per person for these devices is also expected to increase. The energy efficient assessment of IoT devices by IEA 4E Electronic Devices and Networks Annex ("EDNA") [Friedli, 2016] published this year, April 2016, predicts the standby energy consumption of these IoT devices to increase at a 20% rate, reaching 46TWh in the year 2025 (see Figure 1) , which was the Portugal's entire annual electricity consumption in the year 2012.

1.2 Why subthreshold operation?

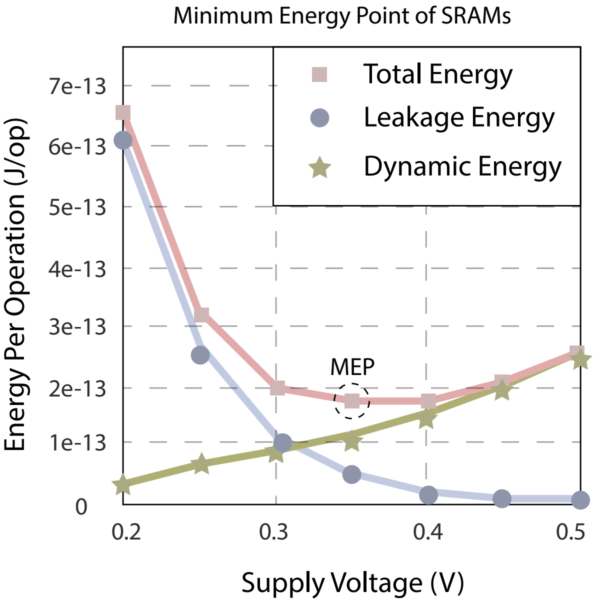


Figure 2: Minimum Energy per Operation of six-transistor SRAM memory bitcell. Source [Banerjee and Calhoun, 2014].

More than 50% of the area in a processor die is taken by the on-chip memory circuit which includes the registers, latches/buffers and cache memory. The memory circuit also has the highest contribution in the leakage energy consumption of the chip. As such, memory becomes the dominant source of energy dissipation compared to other parts of the processor. It dictates the power/energy envelop of the processor design, which in-turn determines the minimum operating voltage of processor and the range of dynamic voltage scaling (DVS) that can be used for power efficiency. Thus, memory design takes central role in designing power/energy efficient processors.

In the ultra-low-power/energy domain, reducing power/energy consumption is the priority for the designers which is achieved by lowering the supply voltage. The dynamic energy consumption of a circuit is proportional to V_{DD}^2, where V_{DD} is the supply voltage of the circuit. Thus decreasing the supply voltage of the circuit gives quadratic reduction in the dynamic energy consumption of a circuit. However, an important consequence of the reduction in the operating voltage is an exponential increase in the delay of the circuit, which results in the increase of the leakage energy consumption of the circuit.

At voltages near sub-threshold range, it has been shown [Calhoun et al., 2005] that the total energy consumption per operation of the circuit reduces to a minimum value . The minimum energy operation of SRAM memory as shown in Figure 2 [Banerjee and Calhoun, 2014], exists in the subthreshold region of operation below 0.4V supply voltage. Operating at this minimum energy per operation point (MEP) can provide 20x decrease in the total energy consumption of the circuit at the cost of 4x larger delay, [Hanson et al., 2006].

Achieving this minimum energy per operation of the circuit has potential to settle the energy consumption concerns of the increasing number of future IoT devices that were mentioned in the previous section.

Furthermore, the growing need for more functionality in these devices demands that the ongoing extreme miniaturization of the transistor dimensions continues. This shrinkage of transistor dimensions has led the industry astray from the Dennard's MOSFET scaling rule, which is based on keeping the same electric field as we scale down to lower technologies. If the Dennard's rule could have been followed, same power density and lower delay upon moving to the lower technology nodes would have been possible. However, the industry has not been able to follow this rule and the minimum operating voltages of the circuits have started to stagnate. A way out would be achieving reliable operation at subthreshold operating voltages.

1.3 Challenges of operating in the subthreshold region

1. **Longer access delay and variation in delay:**
   As the supply voltage in the subthreshold region of operation is below the threshold voltage of the transistor, the circuit operates only on the subthreshold current as the I_{ON} $(V_{GS} = V_{DD})$ and I_{OFF} $(V_{GS} = 0)$ currents. The subthreshold current has an exponential dependence on the supply voltage and on the threshold voltage of the transistor.
   This leads to the following problems:
   - A small decrease in the supply voltage results in significant decrease in the I_{ON} current of the circuit. Due to this weak subthreshold current, it takes longer time for the read bitline capacitance to discharge during the read operation (see Figure 3). The consequent increase in the read access time leads to an increase in the read access timing failures. A comparison of the distribution of the read delay of SRAM memory under 10% threshold voltage variations between above-threshold (1V) and sub-threshold (0.4V) supply voltages is shown in Figure 4. In the subthreshold operation, 52.8% of the simulations failed to generate a bitline differential of greater than 50mV even after 500ns.

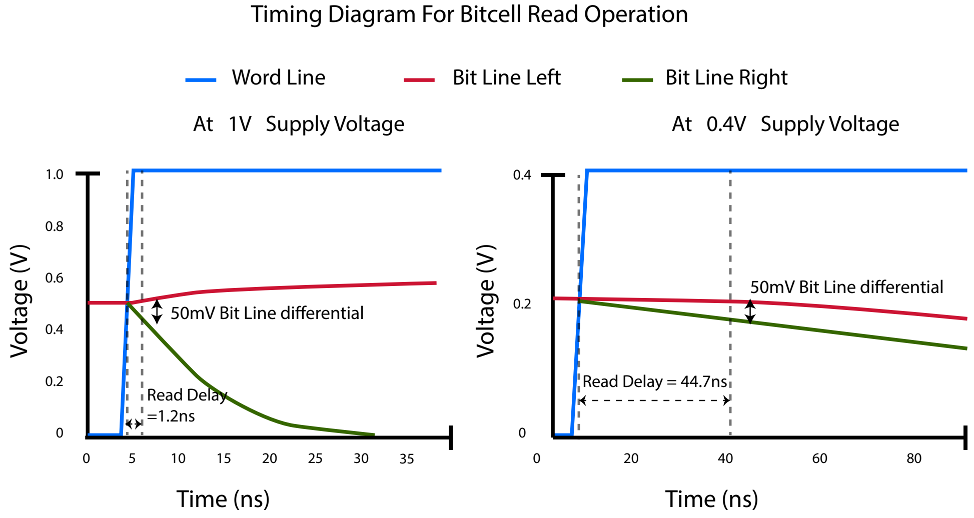


Figure 3: Comparison of delay in the bit-line capacitance discharge at above threshold (1V) and subthreshold (0.4V) supply voltages. At subthreshold voltages, discharging the bitline capacitance is slower because of smaller transistor current.

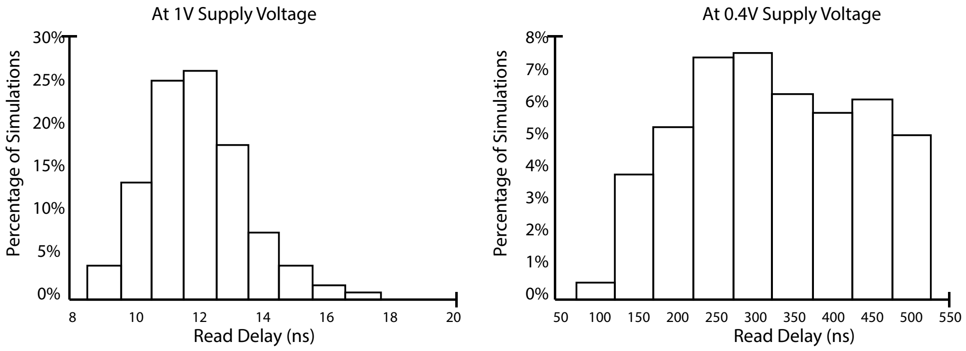


Figure 4: Comparison of read delay of six-transistor SRAM memory bitcell at above threshold (1V) and sub-threshold (0.4V) supply voltages in the presence of 10% threshold voltage variations. The subthreshold operation has larger spread in read delay. Moreover, 52.8% of the simulations failed to discharge the bitline during read operation.

- While operating at the above threshold voltages, the drift current in the transistor acts as the I_{ON} current and dominates the diffusion current I_{OFF}. Thus there exists a clear distinction between the I_{ON} and I_{OFF} current levels. However, at subthreshold supply voltages, the drift current ceases to exist. The distinction between I_{ON} and I_{OFF} currents become very small in magnitude, the Figure 5 shows that the ratio becomes 1/100th of that available at above-threshold voltages. In the case of read operation of the memory, the reduction in I_{ON}/I_{OFF} leads to an increase in the read failures because the difference vanishes between the read-bitline capacitance discharge through the I_{ON} current of the accessed bitcell and the I_{OFF} current from the non-accessed bitcells in the same column of the memory array.

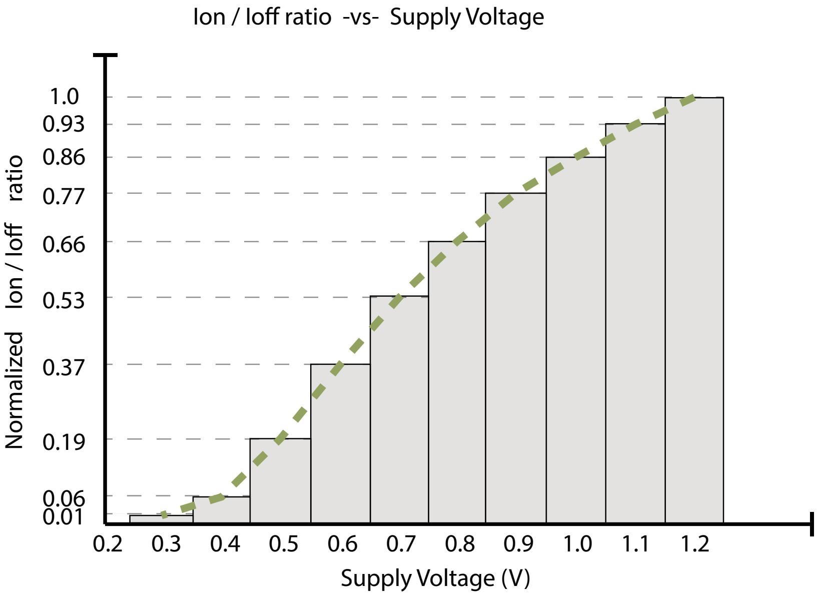


Figure 5: Decrease in the I_{ON}/I_{OFF} ratio as the supply voltage is down scaled to subthreshold region. The I_{ON}/I_{OFF} ratio values are normalized to that at 1.2V supply voltage. At subthreshold supply voltage of 0.3V, the I_{ON}/I_{OFF} is ≈ 1/100th of the value at 1.2V. The decreasing I_{ON}/I_{OFF} is a problem for the SRAM memory because it can lead to the reading a wrong value due to the leakage current from the non-accessed bitcells of the array column.

2. **Higher probability of failure in memory margins:**
   The performance metrics of the SRAM memory being dependent on the current characteristics. They also become exponentially dependent on the threshold voltage variations. The vanishing memory noise margins at ultra-low voltages aggravated with the exponential dependence of I_{ON} on the threshold voltage variation leads to increased failure rates in the read, write and hold operation of the memory bitcells. [Raychowdhury et al., 2005, Calhoun and Chandrakasan, 2006a]

3. **Difficult to achieve minimum energy operation in presence of process variations:**

   - The presence of process variations results in the existence of a distribution of energy consumption per operation rather than a single value as is the case for operation under nominal conditions (absence of process variations). If the energy consumption per operation is assumed to be normally distributed (not necessarily true!) at any arbitrary subthreshold supply voltage, the operating voltage for minimum energy per operation point (MEP) can be taken, as an example, to be the voltage for the minimum $(\mu_{energy} + 3\sigma_{energy})$ which covers 99.7% of the energy values under the distribution. It is obvious that this MEP voltage under process variations is going to be higher than the MEP voltage for nominal conditions [Hanson et al., 2006]. Finding this $(\mu_{energy} + 3\sigma_{energy})$ MEP voltage requires the estimation of energy distribution at all voltage sweep points during SPICE simulation using Monte Carlo simulations. The astronomical increase in the SPICE simulation budget will also increase the cost of IoT devices to unreasonable levels; and the longer time to market will lead to missed opportunities in the fast-moving IoT industry.
   - Resiliency of the memory circuits to process variations can be improved by up-scaling the transistor sizes. The increase in transistor dimensions of the memory increases the dynamic energy consumption and decreases the leakage energy consumption per memory access. The minimum energy per operation point can then shift from a lower MEP energy value at a sub-threshold supply voltage to higher MEP energy value at an above-threshold supply voltage. This goes against the motivation for achieving the energy efficient operation at subthreshold voltages.

1.4 Roadmap of this thesis

1.4.1 Thesis Objectives

In this section, the objectives for the research work done on the sub-threshold memory operation are described. First, we highlight the problem that estimation of the failure probability of the memory margins needs millions of SPICE simulations. A faster estimation methodology is needed to reduce the time and computational cost spent on the analysis and the yield optimization of subthreshold operation of the memory. Then we emphasize on the need to design subthreshold SRAM bitcell topologies not just considering the reliability aspect, which has been the approach taken by the existing proposals, but also finding trade-offs that will reduce the minimum-energy operation of the memory bitcell. Finally we describe the need to explore the potential of e-DRAM gain cells as alternatives to SRAM bitcells so as to achieve the minimum energy operation in the size and energy constrained IoT devices.

1. **Faster estimation of memory failure probability:**
   The problem of finding faster approaches to estimate the memory yield, is common to both the above-threshold and the sub-threshold operation of SRAM memory. The probability of memory margin failure is given by the area within failure regions under the multidimensional probability density function of the threshold voltage variation. When the operating voltage decreases, this failure region in the threshold voltage variation space increases and thereby, the probability of failure in memory margin increases. The traditional approach of Monte Carlo simulations to estimate the integral of density function under failure region costs SPICE simulations proportional to the failure probability which is to be estimated. To estimate a failure probability of 10^{-6}, more than one million Monte Carlo samples are needed so that at least one of the sampled points is a failure point, otherwise the Monte Carlo method fails to provide a failure probability greater than zero. The recent proposed works on faster memory margin failure probability estimation have adopted techniques from statistics literature that provide significant improvement over the limitation of Monte Carlo sampling. The sampling method begin used most widely nowadays is the Importance Sampling method [Doorn et al., 2008]. The two main variants of the Importance Sampling that dominate the literature of estimating memory yield are - Minimum Norm Importance Sampling [Dolecek et al., 2008], and Mixture Importance Sampling [Kanj et al., 2006]. The SPICE simulation cost for these methods is still in tens of thousands when estimating very low failure probability values ($< 10^{-6}$). Moreover, Importance Sampling suffers from significant problems related to its convergence (infinite variance) and inability to scale to higher dimensions. These pitfalls of Monte Carlo and Importance Sampling approaches are discussed in the next chapter.

2. **Novel subthreshold SRAM bitcells**
   The 6T SRAM bitcell is traditionally used in memory caches operating at high supply voltages. The margins (read/write/hold) of this bitcell vanish when supply voltage is scale down to subthreshold voltages below 0.3V. Furthermore, in the presence of process variations, the bitcell loses its capacity to provide correct operations at near-threshold voltages. Achieving minimum-energy and robust operation is an unattainable goal with 6T SRAM bitcell. While, the research on memory design robust to process variations at higher supply voltages has been going on for last many years. The emphasis on achieving reliable memory operation for minimum energy operation at subthreshold voltages is a recent direction in the research. The search for alternatives to the 6T SRAM bitcell has led to proposals of 8T [Verma and Chandrakasan, 2008], 9T [Chang et al., 2012], 10T [Kulkarni and Roy, 2012] and 12T [Chiu et al., 2014] SRAM bitcells achieving superior margins. The larger silicon

area of these subthreshold bitcells compared to 6T SRAM bitcell is a limiting but necessary downside of these proposals. Moreover, these proposals of subthreshold bitcells have till yet mainly focused only on the reliability aspect of subthreshold memory operation. The main motivation for subthreshold memory operation is to achieve minimum energy per operation. However, the minimum-operation-per-operation voltage of these bitcells is likely to reach near threshold region because of their larger transistor count. Therefore, the optimal design of subthreshold bitcells providing reliable memory operation by increasing the transistor count per bitcell, must also provide trade-offs to mitigate the resulting increase in energy per operation.

3. **Alternatives to SRAM bitcells for subthreshold operation**
The emerging IoT devices and Bio-medical wearable devices are limited in the energy available per operations because of the size constraints and hence use small-capacity embedded-battery. The lifetime of these devices thus can be increased by achieving energy-efficiency of the highest order. The current architecture of wearable devices targeted towards low-performance domains such as TeleHealth have small on-chip SRAM memory. For instance, NXP's LPC1102/1104 a 32-bit ARM Cortex M0 MCU has 32kB flash and 8kB SRAM, [NXP,]. Here, the strict size constraints make it impractical to achieve reliable memory operation by the up-sizing of transistor dimensions. The 2T and 3T1D gain cells have larger device density and lower leakage (because of fewer transistors) than SRAM bitcells. 3T1D e-DRAM gain-cell is shown to be capable of achieving access speeds comparable to the 6T SRAM [Liang et al., 2008] at above threshold voltages. Thus, e-DRAM gain cells need to be investigated as an alternative to the SRAM bitcells in caches for low-energy devices.

1.4.2 Thesis contributions

1. **Improved sampling process for Mean-Shift Importance Sampling - *SSFB* and *REEM*:**
The early work on this thesis focused on reducing the SPICE simulation cost of memory failure estimation. We adopted the Mean-Shift Importance Sampling method because of its ease of implementation. This sampling method is dividing into two steps:

    a) Random sampling in the threshold voltage variation space to find the most probable failure point (MPFP).

    b) Shifting the mean of the original probability density function of threshold voltage variation to this new found MPFP. The resulting distribution is then used to estimate the failure probability of memory margins using Importance Sampling approach.

    The proposal SSFB focused on reducing the SPICE simulation cost of step "a" by using radial simulations to reach to the failure boundary in the threshold voltage variation space and followed by the random sampling only within a sub-region of the hyper-sphere surface at the failure boundary. The results for this approach showed 40x reduction in the SPICE simulations to estimate the MPFP compared to the random sampling approach used in previous proposals.
The second proposal REEM reduces the SPICE simulations in both steps "a" and "b". The method maintains an estimate of the failure and non-failure regions in the threshold variation space and guides the sampling process to those regions where the largest increase in the estimates of failure/non-failure region is possible. The method eventually provides an estimate of the failure boundary near the MPFP. The subsequent estimation of the memory failure probability using Importance Sampling at the MPFP can be done using the estimated failure/non-failure regions. The random samples lying in these estimated regions do not need SPICE simulations. The results for the proposed method showed 10x reduction

in the SPICE simulations compared to the Mean-Shift Importance Sampling proposals to estimate the memory margin failure probability.

2. **Gaussian process regression based surrogate models for memory margins using additive kernels:**
The previous proposal REEM and SSFB focused on improving the sampling process of Mean-Shift Importance Sampling method to estimate the failure probability. These methods still require tens of thousands samples and hence are expensive in the number of SPICE simulations needed. We tackled this problem further by building surrogate models of the dynamic margins of the SRAM bitcell using Gaussian process regression. The subthreshold current of the bulk-CMOS transistor has an exponential dependence on its threshold voltage. The resulting variation in the ON/OFF current of the transistor in the presence of threshold voltage variations is thereby highly non-linear. The modeling of this behavior thus demands for regression methods that can capture this non-linear behavior easily from the simulated samples. This is not an easy task with the linear regression class of methods because there is no beforehand knowledge available about the appropriate class of basis vectors (do we need only x, x^2, x^3, etc. for polynomial regression or more complex basis functions $\sqrt{x}$, $\log(x)$, e^x, etc. or even composition of these basis functions $\sqrt{(\log(x))}$ etc.) that can be used to model the non-linearity present in the sub-threshold operation of memory, [McConaghy, 2011]. Gaussian processes regression (GPR) is a non-parametric regression method where the knowledge of these basis vectors is not needed in order to build the surrogate model from the simulation data. The predictions of a model built using GPR are dependent only on a kernel function (typically parametric) that is a co-variance function between any two points on the variable space. The existing proposals for surrogate modeling of memory used universal kernels such as Radial Basis Kernel function (RBF). While these kernel functions have the capacity to model any smooth function from the training observations, this large capacity comes with a trade off of higher out-of-sample error because these universal kernels cannot extrapolate at locations farther from the training samples. We proposed to use additive kernel functions which can extrapolate the margin values from the simulated samples and achieved 32% lower out-of-sample error compared with the Radial-Basis-Function kernel (RBF). An additive kernel function decomposes into the sum of low dimensional kernel functions and hence reduces the dimension of the surrogate model. The low dimensional kernels in our case of SRAM margin modeling were used to model the sensitivity of the margin with respect to the threshold voltage variation in each of the transistors of the SRAM bitcell. The additive kernel thus gave a surrogate model encoded with this information about sensitivity of the margin with respect to threshold voltage variation in individual transistors. The surrogate model built using 1250 SPICE simulations gives predicted failure probability values with accuracy numbers similar to the previous proposals while the reduction in the SPICE simulation cost is between 4x and 23x compared to the previous surrogate modeling proposals and 800x compared to the Monte Carlo method.

3. **10TSD: Sub-threshold Bitcell for Faster Read Access:**
The existing near/sub-threshold SRAM bitcells all increase the transistor count per bitcell for higher stability and reliability than the 6T bitcell. Consequently, there are more number of transistors in the discharge path of these bitcells and thereby, longer access delay. These subthreshold bitcells only find use case in the low-performance energy-constrained domains like wireless sensor networks. In this work, we show a new cell design that can also operate efficiently in mid-performance domains (i.e. it has a wider voltage and frequency range operation). We presents a novel 10T single-ended (Single-transistor-Discharge-path) near-threshold bitcell, 10TSD, that can operate between 2x and 3x the speed of previous 10T

cells in the near-threshold range, while ensuring similar noise margins, plus it reduces both the read '0' and Read '1' dynamic energies by 70% and 30% respectively. The drawbacks are the increase in Read '1' and Read '0' leakage energy which is 17% and 42% respectively when operating at high voltages (i.e. 0.5V) and 13.6% increase in the bitcell area.

4. **Statistical analysis of e-DRAM gain cell operation at subthreshold voltages:** Recent proposals have investigated replacing the flash memory with e-DRAM gain cells with promising results showing increase in energy efficiency. This work examines minimum-energy operation of 2T and 3T1D e-DRAM gain cells as an alternative to SRAM at 32nm technology node with different design points:
   - Up-sizing transistors
   - Using high-threshold voltage transistors
   - Read and write word-line assists
   - Temperature

   First, the work investigates the e-DRAM gain cells without considering the process variations. In order to reduce the SPICE simulations to explore the design-space with above mentioned parameters, kriging meta-models of the e-DRAM read energy, retention time and delay at the MEP are used. Finally, a full-factorial statistical analysis of e-DRAM gain cells is performed in presence of threshold voltage variations to investigate the effect on the mean energy at MEP for the read operation.

1.4.3 Thesis Organization

The structure of the thesis is as follows:

- Chapter 1 provided the motivation for the research problem and lays down the objectives of the thesis work. A summary of thesis contributions were then presented.

- Chapter 2 provides background on the subthreshold memory operation. discusses the pitfalls in Monte-Carlo and Importance Sampling techniques, which are heavily used nowadays to estimate the probability of failure in memory margins.

- Chapter 3 introduces the sampling approaches SSFB and REEM : Both these approaches reduce the simulation cost of the Mean-Shift Importance Sampling method, thereby enabling faster estimation of the memory yield.

- Chapter 4 introduces the surrogate modeling approach using Gaussian process regression: Using surrogate models enables further reduction in the simulation cost of estimating the memory yield. Here we present an additive kernel based approach which is better tuned to SRAMs than a universal kernel such as Radial Basis Kernel functions (RBF) capable of modeling any smooth function. Restricting the capacity of the kernel functions decreases the out-of-sample error, resulting in more accurate surrogate models

- Chapter 5 introduces an 10T subthreshold SRAM bitcell that provides faster read access.

- Chapter 6 provides the statistical analysis of the e-DRAM gain cells 2T and 3T1D as an alternative to SRAM bitcells for size-constrained low-energy domains.

James Hacker: You said you heard it was true!
Bernard Woolley: No, I said it was true that I heard it!
Annie Hacker: I'm sorry to cut into this important discussion, but do you believe it?
James Hacker: I believe I heard it. Oh, about the diamonds. No.
Annie Hacker: Is it impossible?
James Hacker: No, but it's never been officially denied. First rule in politics: never believe anything until it's officially denied.

Yes Minister: Party Games

2 Background, and Related Work

The theoretical lower limit of ≈ 36mV on the operating voltage of a CMOS circuit has been known since 1972 [Swanson and Meindl, 1972]. However, research on subthreshold digital circuits has gained momentum in the recent years, guided by the need to achieve energy efficiency by operating at minimum energy per operation voltage. In this chapter we first provide background to the problem of faster memory yield estimation. The limitations of Monte Carlo method and pitfalls of Importance Sampling method are discussed. Later we provide a brief summary of the subthreshold SRAM bitcell topologies proposed in recent years.

2.1 Memory Failure estimation methods

The first obstacle faced in the design of subthreshold memory design is the diminishing memory read/write margins. The presence of process variations in the current technologies, results in the failure of 6T SRAM bitcell in read, write and hold operations. The process variations lead to significant reduction in the memory margins for its operation even in the above-threshold regions [Wang et al., 2008], which has resulted in the demand for high-yield memory design methodologies. However, when operating at subthreshold voltages below 0.3V supply, the memory margins for the 6T SRAM bitcell vanish [Calhoun and Chandrakasan, 2006b]. Thus, attaining correct memory operation under process variations is the first step towards subthreshold memory design. Consequently, any approach for subthreshold memory design must estimate the margin failure probabilities and then optimize the memory circuit so as to achieve lower failure probabilities which will make the memory operation at subthreshold voltages realistic. However, computing very small failure probabilities of the memory margins with few thousand simulations is not feasible. It is not difficult to estimate that a memory bitcell fails under subthreshold operation with few simulations because at these ultra-low supply voltages it has large failure probabilities. The difficulty is in the yield-optimization phase where it is necessary to evaluate accurately the effect of upsizing transistor dimensions, assist techniques and novel bitcell topologies on the failure probability. As the failure probability of the memory margins decreases, the SPICE simulation budget needed to verify the results also increases.

2.1.1 Challenges to the Monte Carlo method

Monte Carlo method is an approximation method to calculate the integral of a function. If a function "θ" has domain as the sample space of a random variable "x" with probability distribution "$f(x)$", then its integral is the expectation of the $\theta(x)$. In our case of estimating failure probability, the function is the failure indicator function "$I(x)$" where the random variable "x" is the variation in the process parameters. Assuming that the effect of process variations are lumped together as a single variation in the threshold voltage of a transistor with a Gaussian distribution $f(x)$; then in the case of 6T bitcell, the random variable "x" are the variations in the threshold voltages of each of the six transistors with sample space Ω as a subset of R^6. The required failure probability is then the expectation of this failure indicator function:

$$\begin{aligned} \mathrm{Prob.Fail.} &= E_f[I(x)] \\ &= \int_{\Omega} I(x) f(x) dx \end{aligned} \tag{1}$$

The Monte Carlo method generates “M” random samples with probability distribution $f(x)$ from this sample space Ω and provides an estimate of failure probability by averaging the indicator function values evaluated at these samples.

$$\mathrm{Prob.Fail.}_{MonteCarlo} = \frac{\Sigma_i^M I(x_i)}{M} \tag{2}$$

The law of large numbers ensures that the average computed by the Monte Carlo method converges asymptotically to the integral value. Monte Carlo method provides faster convergence because the error in its estimate is inversely proportional to the square root of the number of samples M, against the deterministic numerical methods of integration which have exponential dependence on the dimension of the sample space Ω. In the case of very small failure probabilities, the Monte Carlo approach becomes non-practical when the failure regions in the sample space Ω are at the extreme ends of the tail of probability distribution $f(x)$. In such scenario, we face the following challenges:

1. Sampling of few thousand points under $f(x)$ is not enough to get a failure sample from the tails of $f(x)$. These non-failure sampled points do not contribute to Monte Carlo estimate of failure probability because the indicator function value at these points is “0”. **Can we reach the failure regions at the tails of $f(x)$ when sampling with few thousand simulations?**

2. Every non-failure sampled point is adding to computation cost of SPICE simulations. If all of the sampled points are non-failure samples, we have wasted the computationally expensive SPICE simulations. **Can we reduce the computational overhead of simulating non-failure points?**

It is clear that the designing subthreshold memory will remain computationally expensive and a slow process using just Monte Carlo approach. To overcome this computational bottleneck, the search for faster sampling alternatives has gained momentum and sampling strategies from the statistics literature are being adopted.

2.1.2 Improvements to the Sampling process

Here we present the approaches that have been adopted in the past:

1. **Importance Sampling:**
   Importance Sampling is a sampling technique where estimations about the original probability distribution are made using samples from a different probability distribution. In the case of estimating failure probabilities of the memory read and write margins, we are trying to find an estimate of the area under the region of margin failure events. In this scenario, when the margin failure probability (which is to be estimated) is very small ($< 10^{-6}$), it is impractical to try to generate these rare-event failure samples from the original probability distribution of the memory margins. Importance Sampling can thus be used to generate these samples of failed margins by using a different probability distribution which covers a larger part of the memory margin failure region than the original probability distribution of the memory margin. Another way of describing this is, it is sometimes impossible to get samples from the failure regions, which exist at the extreme ends of the tail of the original probability distribution, with a small simulation budget of few thousand simulations. It is, then, rather beneficial to sample from a distribution whose tails cover a larger failure region as shown in Figure 6 below.

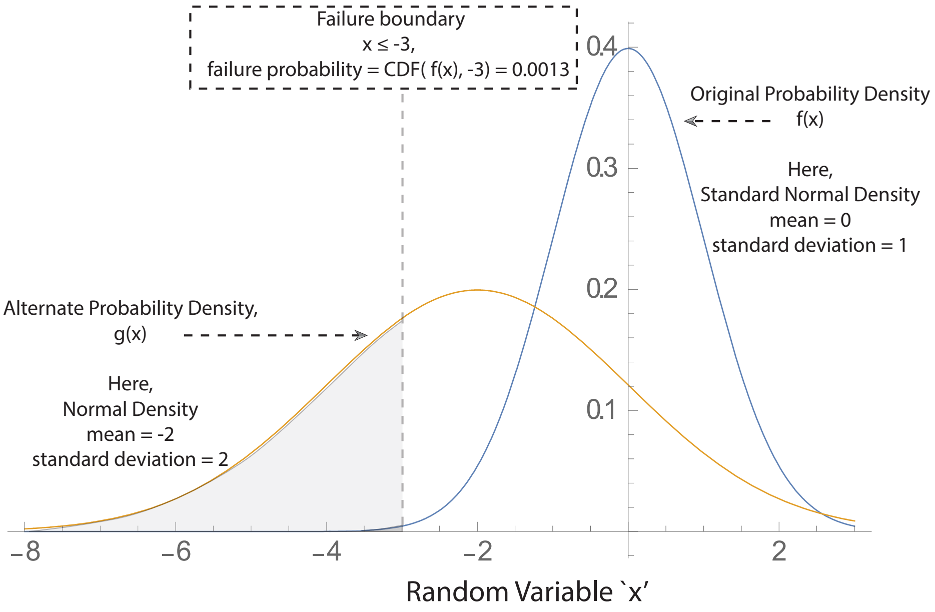


Figure 6: Alternate distribution for sampling more failure points

Since the samples generated are not from the original probability distribution $f(x)$ but from a different distribution $g(x)$, it is necessary to "un-bias" them so that the failure probabilities under the original distribution can be calculated. This is achieved by choosing weights for each individual sample proportional to the inverse of its relative likelihood.

$$W(x) = \frac{f(x)}{g(x)} \tag{3}$$

Thus, the samples which are less likely to get sampled under the $f(x)$ than under $g(x)$, are given smaller weights. This is the case for the failure points sampled under the distribution $g(x)$. They are more likely to be sampled under $g(x)$ than under the original probability distribution $f(x)$ of the memory margins and so must be given smaller weights, otherwise failure probability will be over-estimated. The equation 1 in that case is updated as follows

$$\begin{aligned} \text{Prob.Fail.} = E_g[I(x)] &= \int_\Omega I(x)\frac{f(x)}{g(x)}g(x)dx \\ &= \int_\Omega I(x)W(x)g(x)dx \end{aligned} \tag{4}$$

The Importance Sampling then gives an estimate of the failure probability by the weighted average of the failure indicator function $I(x)$ at the M_g number of points sampled under the distribution $g(x)$.

$$\text{Prob.Fail.}_{IS} = \frac{\Sigma_i^{M_g} I(x_i)W(x_i)}{M_g} \tag{5}$$

The selection of the distribution $g(x)$ is not straightforward. To illustrate this, consider the original distribution $f(x)$ to be the standard normal distribution $N_x(0,1)$ with failure region as $x : x \leqslant -1$. The failure probability in this case is the area of the f(x) under the failure region, i.e,

$$\text{Prob.Fail.} = \text{CDF}(N_x(0,1), x = -1) = 0.16 \tag{6}$$

The optimal Importance Sampling distribution is the one with zero variance, which assigns weight $\text{CDF}(N_x(0,1), x = -1)$ to every point x.

$$\begin{aligned} \text{ProbFail} = & \frac{\Sigma_x I(x)W(x)}{N} \\ & = I(x)W(x) \ // \ \because \text{same weight for all x} \\ & = \frac{I(x)f(x)}{g(x)} \\ = & \frac{I(x)N_x(0,1)}{g(x)} \ // \ \because f(x) = N_x(0,1) \end{aligned} \tag{7}$$

$$\begin{aligned} \implies g_{optimal}(x) = \frac{I(x)N_x(0,1)}{\text{ProbFail}} &= \frac{I(x)N_x(0,1)}{\text{CDF}(N(0,1), -1)} \\ &= \frac{I(x)N_x(0,1)}{0.16} \end{aligned} \tag{8}$$

Thus the optimal distribution for Importance Sampling in this example is,

$$g_{optimal}(x) = \begin{cases} \frac{N_x(0,1)}{0.16} & \text{if } x \leqslant -1, \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

However, we have initially no beforehand knowledge of the failure region and hence do not know that all points, x, with $x \leqslant -1$ are failure points. So we cannot use it to estimate the optimal Importance Sampling distribution. If we had this knowledge, we would not be needing Importance Sampling, or even Monte Carlo estimates for that matter, because the failure probability is then simply given by equation 6.

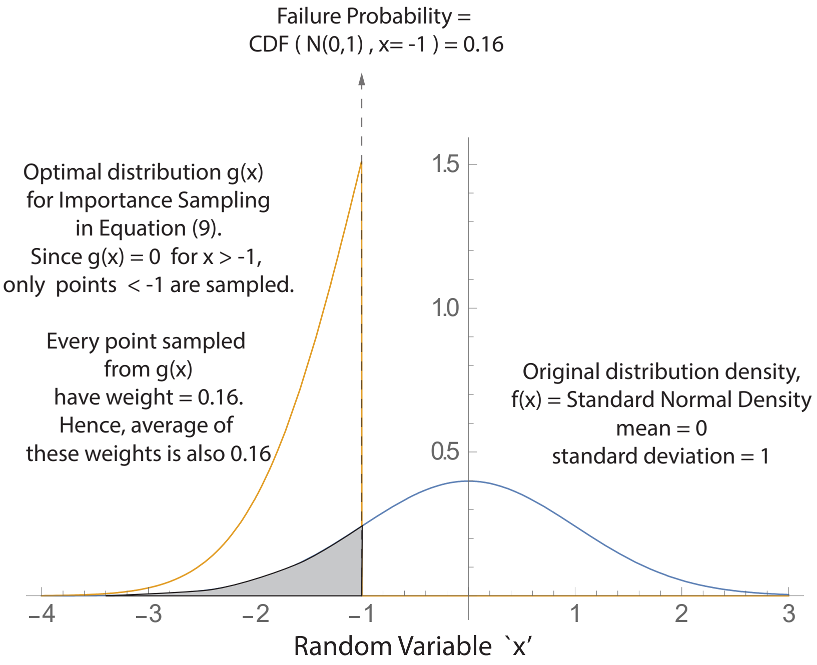


Figure 7: Optimal Importance Sampling distribution for the example.

Let us estimate this failure probability by using $g(x)$ as a discrete uniform distribution with sample space of equally spaced M points in the set $[-3, 3]$.

$$\begin{aligned} I(x)W(x) = \frac{I(x)f(x)}{g(x)} &= \frac{I(x)N_x(0,1)}{U_x(M)} \\ &= \frac{I(x)N_x(0,1)}{\frac{1}{M}} \\ &= I(x) * M * N_x(0,1) \end{aligned} \tag{10}$$

$$M \geqslant \frac{1}{\inf(N_{x\in(-3,3)}(0,1))} \approx 225 \implies W(x) \geqslant 1$$

A sample space of more than 225 points for the discrete uniform distribution under [-3,3] assigns weights greater than one to all the samples. If all the sampled points are failure points (a rare event in case of low failure probability, but nevertheless it can still happen!), then the failure probability estimated from Importance Sampling is greater than one! **Meaningless** result as "probability" is a positive number $\leqslant 1$. Furthermore, with constraint $M \leqslant 225$, lets consider the effect of increase/decrease in the size of sample space M on the Importance Sampling estimate. Increasing M decreases the probability mass function of the uniform distribution. For a fixed M, the region under the original distribution function $f(x)$, is divided into two disjoint sub-regions:

a) Region-I: $\frac{f(x)}{g(x)} \geqslant 1$, for these $W(x) \geqslant 1$

b) Region-II: $\frac{f(x)}{g(x)} < 1$, for these $W(x) < 1$

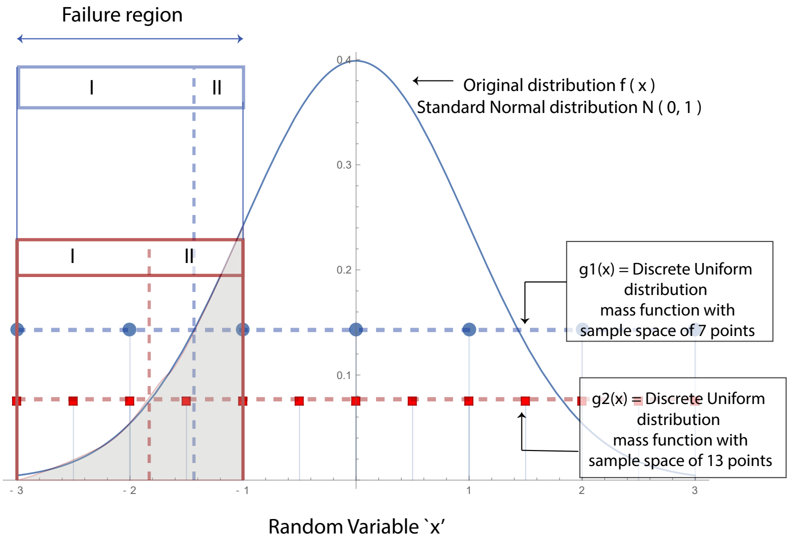


Figure 8: The likelihood ratio $f(x)/g(x)$ partitions the failure region into two disjoint sub-regions, one where samples are assigned Importance Sampling weights less than one (Region-I) and the other where weights of the samples are greater than one (Region-II). Increasing the sampling space of discrete uniform distribution increases the area under Region-II

Since, the sampling distribution $g(x)$ is discrete uniform distribution, the relative proportions of samples (M_I, M_{II}, where $M_I + M_{II} \leqslant M$) lying inside the two region are proportional to the respective lengths (L_I, L_{II}) of these regions.

$$\frac{M_{II}}{M_I} = \frac{L_{II}}{L_I} \tag{11}$$

and the estimate for failure probability is broken down into two parts

$$\begin{aligned} \mathrm{ProbFail} &= \frac{\Sigma M_I * N_{x\in I}(0,1) + M_{II} * N_{x\in II}(0,1)}{M} \\ &= \frac{M_I}{M} * \Sigma N_{x\in I}(0,1) + \frac{M_{II}}{M} * \Sigma N_{x\in II}(0,1) \\ \leqslant \frac{L_I}{L_I + L_{II}} &* \Sigma N_{x\in I}(0,1) + \frac{L_{II}}{L_I + L_{II}} * \Sigma N_{x\in II}(0,1) \\ &= S_I + S_{II} \end{aligned} \tag{12}$$

The proportion of the contribution of regions I and II to the failure probability estimate by the sums S_I and S_{II} is thus proportional to the size of the regions I and II,

$$\frac{S_{II}}{S_I} = \frac{L_{II} * \Sigma N_{x\in II}(0,1)}{L_I * \Sigma N_{x\in I}(0,1)} \tag{13}$$

where $N_{x\in II}(0,1) > N_{x\in I}(0,1)$. Upon increasing the size of sample space M, L_{II} increases and L_I decreases. Thereby the contribution of S_{II} increases while that of S_I decreases. Since the sum S_{II} is over the region II with weights $\geqslant 1$, the estimate for failure probability increases and will eventually become $\geqslant \mathrm{CDF}(N(0,1),-1)$ with larger sample size. Thus we reach a very counter-intuitive result, **increasing the size of sample space under the discrete uniform distribution as $g(x)$ does not necessarily increase the accuracy of Importance Sampling estimate**. Contrast this with the Monte-Carlo estimate, whose variance decreases proportional to $\frac{1}{\sqrt{M}}$. The likelihood of a sample from a continuous uniform distribution is dependent only the 1/length of the sampling space (for instance, sampling space is an interval in one-dimensional case) and not on the number of samples. Thus we would have an increase in the region II under continuous uniform distribution when we increase the sampling interval from say, $[-3\sigma, 3\sigma]$ to $[-6\sigma, 6\sigma]$.

While the Importance Sampling has the potential to provide a zero variance estimate when the optimal distribution $g_{optimal}(x)$ is used. In practical usage, the estimate for failure probability given by Importance Sampling can be a meaningless value if due consideration is not given when choosing the sampling distribution $g(x)$.
Furthermore, the accuracy of Importance Sampling does not scale as the dimension of the sample space Ω increases. In an N-dimensional Ω space, with $f_i(x)$ and $g_i(x)$ as the marginal distributions of $f(x)$ and $g(x)$ in dimension i and assuming independence, the Importance Sampling weight function dependent on the inverse likelihood ratio is,

$$W(x) = \frac{f(x)}{g(x)} = \prod_{i=1}^{N} f_i(x)/g_i(x) \tag{14}$$

Even if the likelihood ratio $f_i(x)/g_i(x)$ is reasonably small but > 1, the product increases in exponential to the increasing values of N. This results in assigning larger weights to the sample points which are less likely to be sampled under $f(x)$ and as a result decrease accuracy of the failure probability estimate. [Doorn et al., 2008] investigated Importance Sampling for SRAM static noise margins by using a large variance distribution for alternate distribution $g(x)$ and on comparison with the extrapolation of Monte Carlo estimate, found Importance Sampling to be more accurate.

2. **Mixture Importance Sampling:**
Mixture Importance Sampling is the extension of the Importance Sampling technique where random variables are sampled from more than one distribution. The resulting sampling distribution can then be expressed as a linear combination of the composing distributions, $h_1(x), ..., h_n(x)$.

$$g(x) = \sum_{i=1}^{n} \lambda_i h_i(x) \tag{15}$$

where, $\lambda_i > 0$ and $\sum \lambda_i = 1$ In this case, random samples are generated by the distribution h_i with probability λ_i. For example, a mixture distribution can be created as the linear combination of the original distribution $f(x)$ and another distribution $h(x)$,

$$g(x) = \lambda f(x) + (1-\lambda)h(x) \tag{16}$$

where $0 < \lambda < 1$. In this scenario, random variables are sampled with probability λ from the original distribution $f(x)$ and with probability $1-\lambda$ from the other distribution $h(x)$. The likelihood ratio, $\frac{f(x)}{g(x)}$, then is bounded above,

$$\begin{aligned} W(x) = \frac{f(x)}{g(x)} &= \frac{f(x)}{\lambda f(x) + (1-\lambda)h(x)} \\ &= \frac{1}{\lambda + (1-\lambda)\frac{h(x)}{f(x)}} \\ &< \frac{1}{\lambda} \\ // \because \frac{h(x)}{f(x)} &> 0 \text{ and } \lambda < 1 \end{aligned} \tag{17}$$

The existence of this upper bound restricts the growth function of the Mixture Importance Sampling weights as the dimension of Ω increases compared to the traditional Importance

Sampling as discussed in the previous Equation 14. However, this only holds if the mixture distribution is composed of whole distributions $f(x)$ and $h(x)$ and not their marginal distributions $f_j(x)$ and $h_j(x)$ for a dimension j [Hesterberg, 2003]. Otherwise, the upper bound of weights for the product of marginal distributions increases with the dimension N of sample space Ω as $1/\lambda^N$. The Mixture Importance Sampling adds to the complexity of the estimation process as now optimal λ_i values are needed along with finding the appropriate sampling distributions $h_i(x)$. The application of Mixture Importance Sampling to the analysis of SRAM failure events was explored in [Kanj et al., 2006] and was later used to study of impact NBTI and PBTI in SRAM bitcells in [Bansal et al., 2009] and gate leakage current in PD/SOI SRAM bitcells in [Kanj et al., 2007].

3. **Minimum Norm Importance Sampling:**

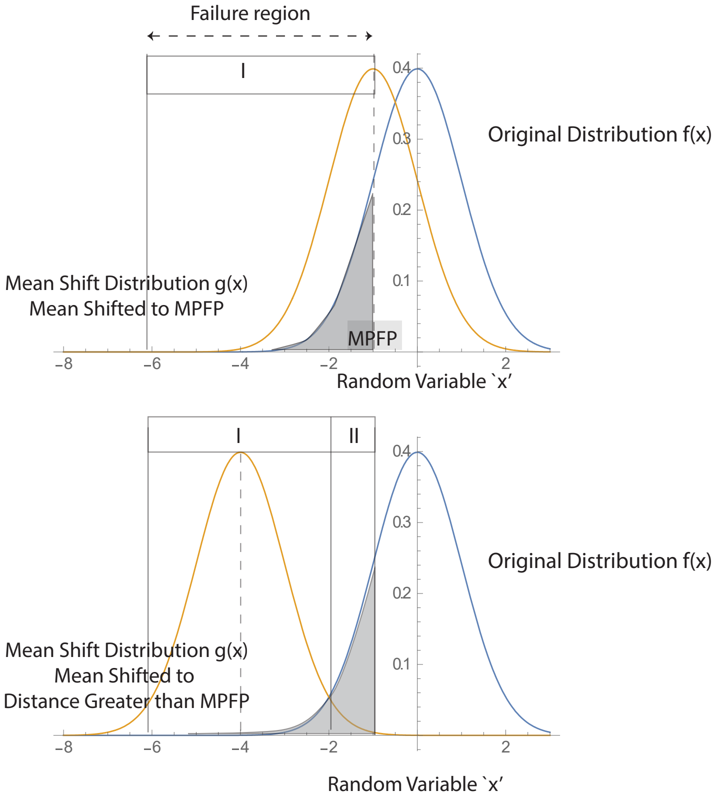


Figure 9: Minimum norm Importance Sampling. Shifting mean away from MPFP increases the region where samples are assigned Importance Sampling weights greater than one.

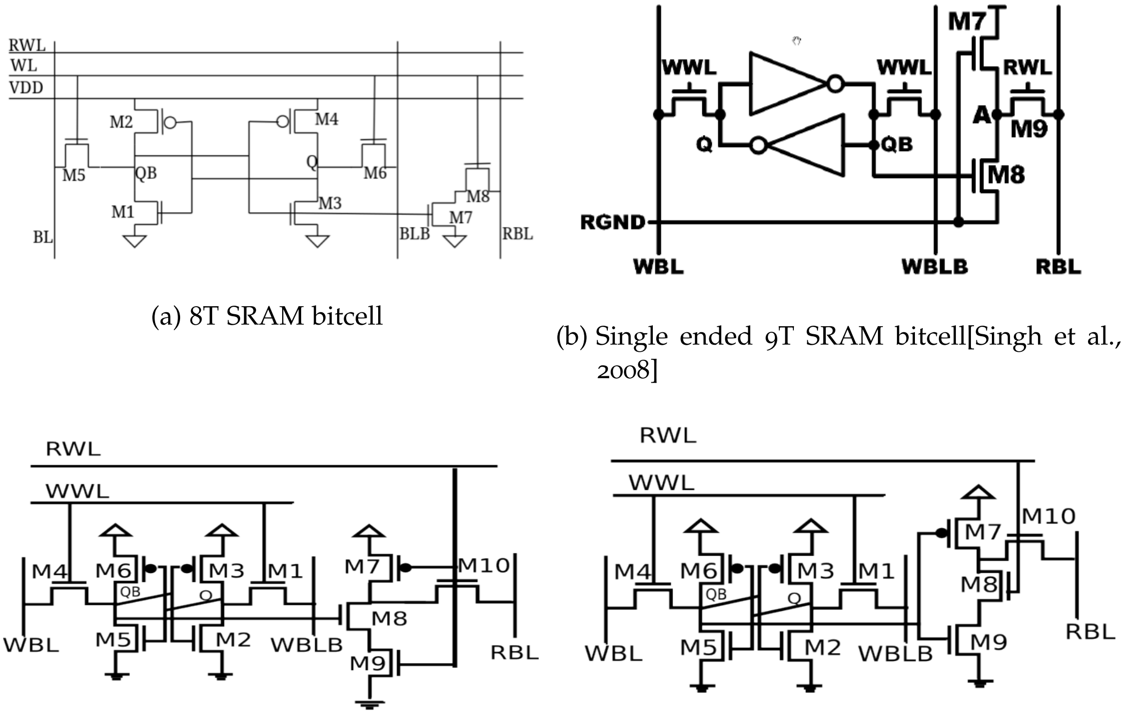
Here the alternate probability distribution $g(x)$ is the original probability distribution $f(x)$ with its mean shifted to the most probable failure point on the sample space Ω. The method, applied to the case of SRAM memory, was justified in [Dolecek et al., 2008] based on an insight from the theory of large deviations: *When a rare event happens, it happens in the most likely manner and hence the probability of this rare event can be estimated from that of this most likely aspect of it*. In the case of SRAM memory, under the assumption of threshold voltage variations being normally distributed as $f(x)$, the most probable failure point (MPFP) is the minimum norm failure point in the threshold voltage variation space. Hence, the method proposes to shift the mean of $f(x)$ to the MPFP. Reviewing our previous discussion on Importance Sampling, we can also provide *another reason* for choosing MPFP as the shift vector for the mean. If we shift the mean of the original distribution $f(x)$ to location farther from MPFP, there is an increase in the region (the Region II) of the sample space where the Importance Sampling weight $(f(x)/g(x))$ assigned to the samples is greater than one, (Figure 9).

As we discussed earlier, increase in this region would increase the variance of the Importance Sampling estimate and can even give a meaningless estimate for failure probability of greater than one. On the other hand if the shift in the mean of $f(x)$ is too small, we may end up with no failure samples again as is the case with Monte Carlo simulation. Hence, the ideal shift to the mean of $f(x)$ is to the minimum norm failure point which is the MPFP. Since the first introduction of minimum norm (MPFP) Importance Sampling in [Dolecek et al., 2008], efforts have been made to speed up the estimation of MPFP, such as [Qazi et al., 2010] augmented it with spherical sampling for estimating SRAM timing failures. [Hagiwara et al., 2010] added the steps of incremental hypersphere sampling (IHS) to search for failure regions followed by decremental hypersphere sampling (DHS) to locate MPFP within those failure regions. [Kida et al., 2012] in their consecutive mean-shift method iterated over small shifts in the mean to estimate the MPFP.

2.2 Subthreshold SRAM bitcells

The sub-threshold SRAM bit cells proposed in recent years have adopted one or more of the following upgrades over the traditional 6T bit cell topology.

- To provide a discharge path for the read bitlines, isolated from the internal storage nodes.



(a) 8T SRAM bitcell

(b) Single ended 9T SRAM bitcell[Singh et al., 2008]

(c) 10T-Kim single ended buffer based SRAM bitcell [Kim et al., 2007a].

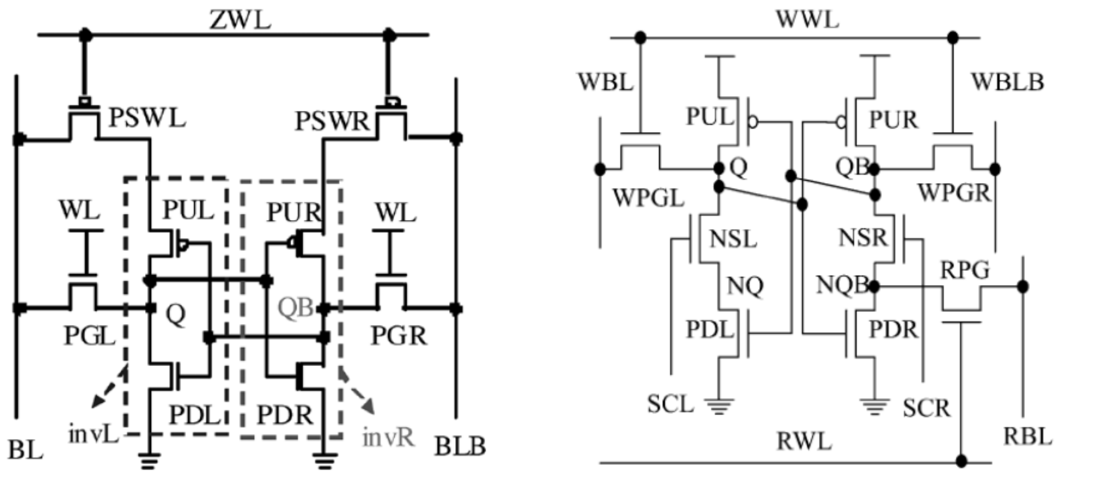
(d) 10T-Calhoun buffer based SRAM bitcell [Calhoun and Chandrakasan, 2006a]

Figure 10: Buffer based subthreshold SRAM bitcells

E.g. the traditional 8T bitcell [Verma and Chandrakasan, 2008], the subthreshold 8T bitcell [Kim et al., 2007b], single ended 9T bitcell [Singh et al., 2008] and the 10T bitcells [Kim et al., 2011, Calhoun and Chandrakasan, 2006a]. These bitcells provide a buffer that reads the value in storage node and the read bitline discharge path is through that buffer. The separate read and write paths with different word-lines also enables the sizing of the transistors in one path to be done independently of the other. This is not feasible in the 6T bitcell case, upsizing the width of the access transistors to increase the write margin decreases its read margin. Larger bitcell area and slower read access are the main issues with these bitcells. The topologies of these bitcells is shown in Figure 10

- To provide a pseudo storage node by adding an extra pull-up or pull-down transistor to the inverter structure.

The bitline discharge path goes through this pseudo-storage node instead of the actual storage node. Thus, the bitline noise does not interfere with the value stored in the actual storage nodes. The idea is same as above to reduce the current influx from the bitlines into the storage nodes. E.g. the 8T and 9T subthreshold bitcell proposed in [Chang et al., 2011, Chang et al., 2010]. These bit cells have the read bitline connected to a pseudo storage



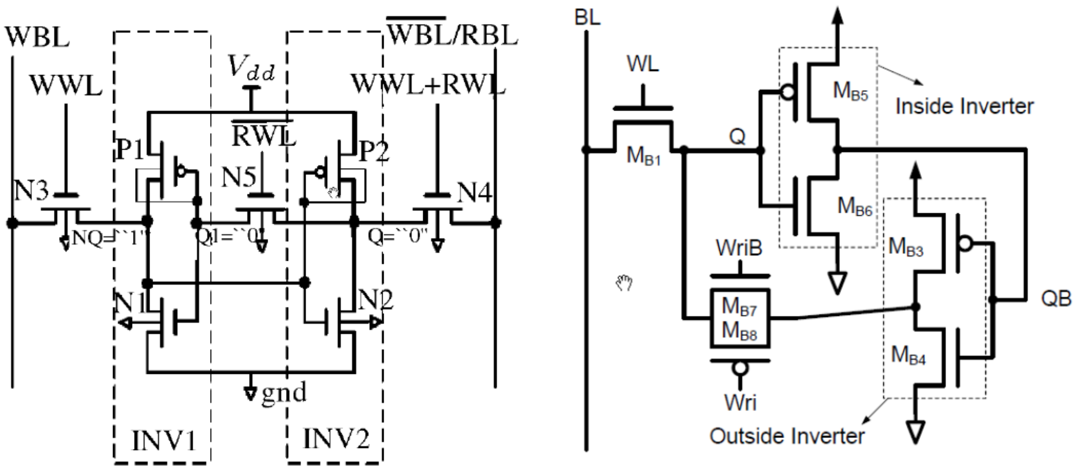
(a) Subthreshold 8T SRAM bitcell [Chang et al., 2011]

(b) Subthreshold 9T SRAM bitcell [Chang et al., 2010]

Figure 11: Subthreshold SRAM bitcell with pseudo-storage nodes

node which is disconnected from the internal storage node during read operation. The main drawback of these bitcells is the modification of the inverter threshold due to the addition of the extra transistors in the inverter structure. The topologies of these bitcells is shown in Figure 11

- To provide a feedback cut-off mechanism between the two inverters.



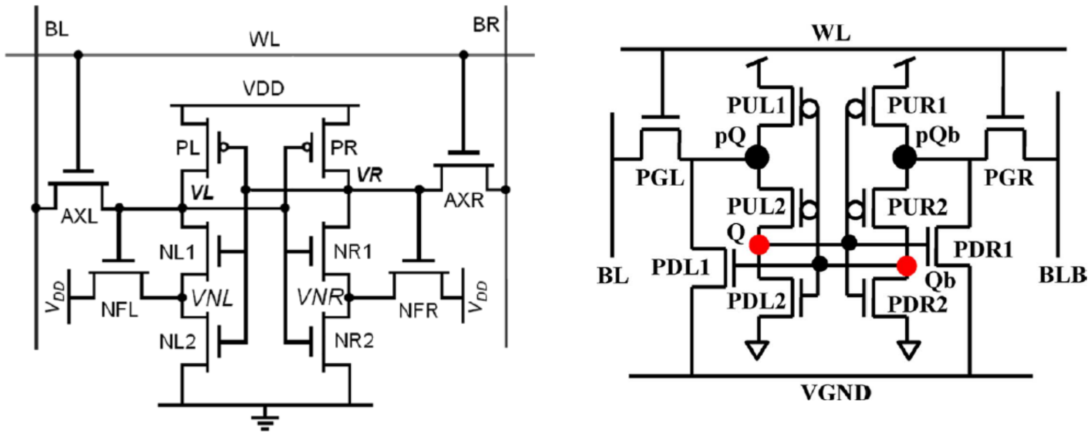
(a) 7T Subthreshold SRAM bitcell [Chang et al., 2012]

(b) 7T Subthreshold SRAM bitcell [Singh et al., 2008]

Figure 12: Subthreshold SRAM bitcells with feedback cut-off mechanism during read or write.

Some proposed subthreshold bitcells break the feedback to increase the write margin of the memory. The write ability in this case is only dependent on the strength of the pass transistors. There also exist proposed bitcells which break the feedback during the read phase. The objective here is to isolate the inverters during read operation so that if any of one of the inverters flips during read, the other storage node retains it original value. If the transistor controlling the feedback in these bitcells has high leakage current, it will not be able to effectively stop the feedback. The 7T bit cell [BAI et al., 2011] cuts off the feedback between the inverters when the cell is being read. On the other hand, the 7T bit cell [Singh et al., 2008] cuts this feedback when the cell is being written.The 9T bitcell proposed in [Chang et al., 2012] tries to increase the write-ability of the cell by breaking the feedback between the inverter loop. The topologies of these bitcell is shown in Figure 12.

- To provide a hysteresis effect to the inverter state transitions to force a higher threshold voltage in the "0" to "1" logic state transition. Thus, the threshold voltage of the inverter storing "0" is higher than the other and it requires a larger voltage drop on the node storing "1" to flip a node storing "0". eg. Schmitt Trigger based bitcell [Kulkarni and Roy, 2012]. Another example of this approach is the 10T-PPN [Lo and Huang, 2011] bit cell. The topologies of these bitcell is shown in Figure 13



(a) 10T-schmitt subthreshold SRAM bitcell [Kulkarni and Roy, 2012]

(b) 10T-PPN subthreshold SRAM bitcell [Lo and Huang, 2011]

Figure 13: Subthreshold SRAM bitcell with hysteresis effect.

Calvin (writing, after being asked to explain Newton's First Law of Motion "in his own words"): *Yakka foob mog. Grug pubbawup zink wattoom gazork. Chumble spuzz. I love loopholes.*

Calvin and Hobbes, Bill Watterson

3 Advances to the Sampling Process for Estimating Memory Failure Probability

3.1 Introduction

This chapter introduces sampling methods to reduce the SPICE simulation cost for finding the Minimum-Norm-Failure-sample, MPFP (proposal SSFB) and Importance Sampling at MPFP (proposal REEM). *This reduction in the SPICE simulation cost is necessary because a memory designer does not have the luxury to spend hours and/or days on the design exploration of a single memory bitcell to achieve higher yields at subthreshold operating voltages. Nevertheless, the design of robust subthreshold memories capable of achieving high yields in the presence of process variations is of prime interest today.*

The failure probability of the memory is estimated by integrating the probability density of process variations in the transistors of a memory bitcell over the failure region. Here we assume that the effects of all the process variation parameters are lumped as a variation in the threshold voltage of a transistor. Monte Carlo methods are typically used to compute this integral whose accuracy varies with number of samples(N) as $\propto 1/\sqrt{N}$. For instance, to achieve a 1% failure rate for a 1MB SRAM array, the required failure probability for a single bitcell is $\approx 1.2 * 10^{-9}$. For Monte Carlo method, the number of required simulations is inversely proportional to the failure probability to be estimated, which implies that for this 1MB SRAM array $\geqslant 10^9$ simulations are needed. The Monte Carlo method is not practical when computing *low probability failures* over high dimensional space as most of the random samples do not lie in the failure region. With Monte-Carlo we face two main problems which has led to a growing interest in other statistical sampling methods, such as Importance Sampling (IS):

1. If the area to be integrated (i.e. failure area when estimating failure probability) is very small; then, many of the samples, obtained by random sampling, will lie outside of that area and will have to be rejected. Only some samples will be inside the failure area that has to be estimated. This will decrease the accuracy of the Monte-Carlo estimate of failure probability as the sum is now taken over a smaller set of samples.
   Can we reduce the number of rejected *samples?*

2. If the Failure-Indicator function is expensive to compute (SPICE simulations are time consuming) the computation of all those *rejected* samples will only add to the simulation count without contributing to the estimation of failure probability.
   Can we instead extract some knowledge about Failure/Non-Failure region from these non-fail samples instead of rejecting them?

The methods, SSFB and REEM, introduced in this chapter assume that the distribution of memory margin failure samples in the threshold voltage variation space is a monotonic function [Khalil et al., 2008]: once a memory bitcell fails for a certain threshold voltage variation value, it continues to fail for larger variation values also. That is, there is no return point to normal operation beyond any failure sample for higher threshold voltage variation. SSFB uses this property to decrease the estimate of failure range at a radial distance in the threshold voltage variation space. In REEM, we exploit this property to classify the failure/non-failure regions with a fewer number of simulations.

This chapter is structured as follows: Section 3.2 briefly describes the previously proposed improvements to the Minimum-Norm Importance Sampling method which will be later compared with our proposals. Section 3.3 describes our first proposal SSFB which estimates MPFP by simulating random samples only near the failure boundary. Section 3.4 describes the REEM method which provides further reduction in SPICE simulations. Finally, section 3.5 concludes this chapter.

3.2 Related Work

In Importance Sampling method, the accuracy of the failure-probability estimator is dependent on the alternate sampling distribution. For this, Minimum Norm Importance Sampling method shifts the original distribution to the Minimum-Norm failure sample (which is the Most-Probable-Failure-sample under normal distribution) with same variation in the distribution. This Most-Probable-Failure-sample (MPFP), obviously, lies on the failure boundary and is the closest failure sample to the origin. Next, we describe the three methods for faster estimation of the MPFP sample using random sampling.

3.2.1 Incremental Hypersphere Sampling Decremental Hypersphere Sampling [Hagiwara et al., 2010]:

This method starts with Incremental-Hypersphere-Sampling (IHS). In this step, failure samples are searched in an annular region within two hyper-spheres. If none of the samples result in failure then radii of the two hyper-spheres is increased by 1σ. Once a failure sample is found, the IHS is then followed by Decremental-Hypersphere-Sampling (DHS) in which radii of the two hyper-spheres 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 within these quadrants only, till an estimate for the minimum-norm sample is found. The illustration of the method for a two-dimensional case is shown in Figure 14. It should be noted here that annular region with large radii hyper-spheres is needed to provide failure samples in the case of very low failure probabilities. The volume of this annular region increases exponentially with dimensions of the sample space (the number of process variation parameters, for example, six threshold voltage variations for 6T SRAM bitcell) which will reduce the sampling density in this region during DHS step, where it is necessary to find the MPFP accurately.

3.2.2 Consecutive Mean-shift [Kida et al., 2012]:

This method addresses the problem of limited sampling density at large radii by using small iterative shifts in the mean of the hyper-spheres with a smaller radii centered at the last found MPFP estimate. It consists of three steps (with illustration shown in Figure 15):

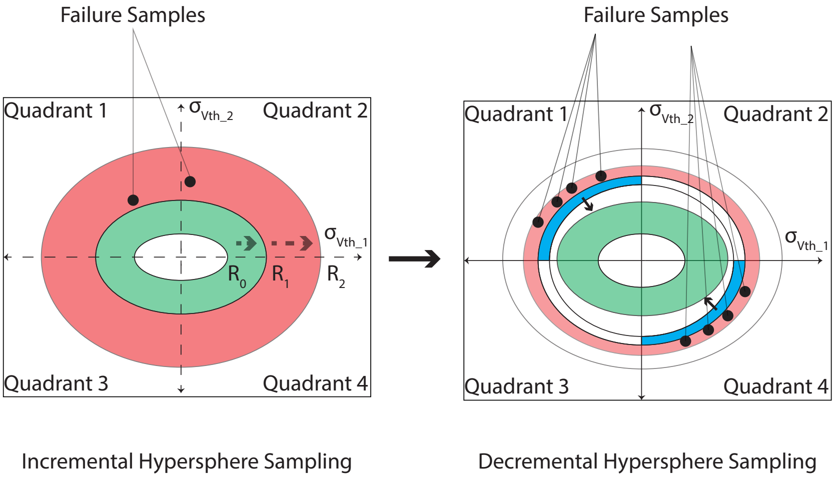


Figure 14: The Incremental-Hypersphere-Sampling (IHS) followed by Decremental-Hypersphere-Sampling (DHS) method [Hagiwara et al., 2010]. The volume of the annular region increases with its radii, which means sample density decreases when failure regions are far away. The method reduces the volume of annular region to search by finding important quadrants during DHS step.

1. Sampling from extended hyper-sphere
   The radius of the Hypersphere centered at origin is increased in steps of 1σ till a failure sample is found. The minimum norm sample among the failure samples is chosen as the initial estimate of MPFP.

2. Sampling from Mean-Shifted hyper-sphere
   Sampling is then done within a hypersphere centered at the initial estimate of MPFP. The estimate of MPFP is updated to the minimum-norm failure sample among the failure samples of the current iteration.

3. Sampling from consecutively mean-shifted IS
   Next sampling is done within a hypersphere centered at the last found estimate of MPFP with small radius to increase the simulation density. This step is iterated till centers of two consecutive hyper-spheres are within a distance of 0.01σ.

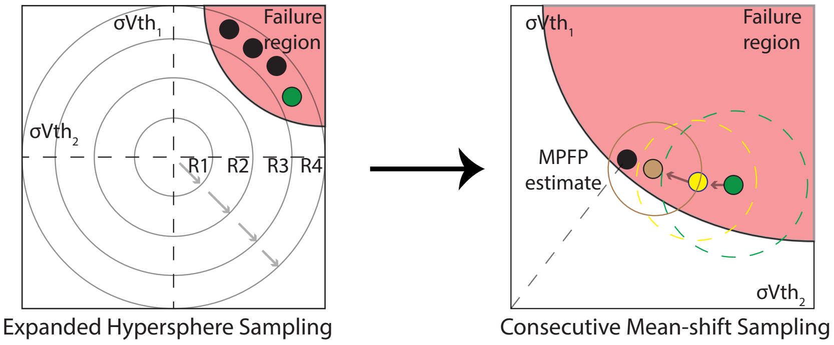


Figure 15: The consecutive mean-shift method [Kida et al., 2012]. After finding failure samples in the initial hypersphere sampling step, the method then samples within hypersphere of smaller radius centered at the last found failure sample with smallest norm. The figure illustrates the convergence of these smallest norm samples among the failure samples to the MPFP estimate.

3.2.3 Sequential Importance Sampling (Particle Filters) [Katayama et al., 2010]:

This method is different from the previously discussed methods in that it does not shift the original distribution to the MPFP. Rather it tries to estimate the optimal distribution for Importance Sampling, $g_{optimal}(x)$ (an example of which was discussed in the Background chapter) by sequentially updating the alternate distribution $g(x)$. Based on the idea of using Bayesian-Filters, the optimal failure distribution $g_{optimal}(x)$ is estimated by shifting particles (random samples) in the process variation space to track the failure regions in three stages (illustrated in Figure 16):

1. *Prediction:* We start with N number of particles (random samples) sampled under alternate distribution $g(x)$ with current position of the samples at locations $X_1, X_2, ..., X_N$. For each particle, based on its current position, the next position is predicted by taking a random sample from a normal distribution $p(x)$ (any other known distribution can be used here for which the likelihood of a random sample can be easily calculated) with mean at the current position.

2. *Measurement:* Likelihood of a particle to lie at the position x after prediction from the distribution $p(x)$ is used to update the Importance Sampling weights which then provide the Importance Sampling estimate of $g_{optimal}$ from these particle locations. Further, for the re-sampling stage, particles are assigned weights which are the likelihood of their locations under the estimated $g_{optimal}$.

3. *Re-sampling:* Particles with higher weights are replicated proportional to their weights and particles with lower weights are eliminated. The process is then repeated by again predicting new locations for the particles.

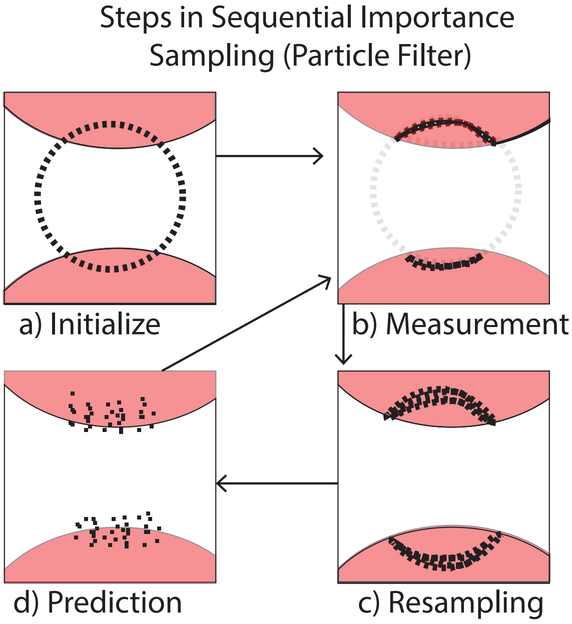


Figure 16: Sequential Importance Sampling (SIS) method [Katayama et al., 2010] shifts the location of each particle to a random location. The non-failure particles are eliminated and particles with higher likelihood under the estimated optimal IS distribution are replicated.

The particle filter approach suffers from the problem of "particle deprivation". This happens when the predicted locations for large number of particle are in non-failure regions. In that case, they are eliminated during re-sampling stage, in the worst case wiping out all the particles. The few remaining particles in the failure regions are replicated. This increases the variance of the estimated failure probability.

The above 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 volume of the hypersphere increases exponentially with dimensions (number of process variation parameters) and hence the samples size (number of SPICE simulations) must also increase to compensate for decreasing sample density.

Our proposals SSFB and REEM are different from these previous approaches in that, with SSFB we do not randomly sample within annular regions between hyper-spheres rather only on the the surface of the hyper-spheres which gives higher sample density for same number of random samples. Further, we do not randomly sample farther from the failure boundary, instead effort is made to reach failure boundary through steps in radial directions from a chosen failure sample and hypersphere surface random sampling is done only upon reaching the failure boundary. In the case of REEM, not *all random samples in the Importance Sampling stage have to be SPICE simulated. Thus Importance Sampling can be done with higher sample count for larger accuracy without increasing the burden of SPICE simulations.*

3.3 SSFB: Simulating Samples near Failure Boundary

In SSFB, we describe the problem in terms of spherical coordinates for hypersphere for ease of implementation. The spherical coordinates are:

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 $\theta_i \in [0, \pi]$

The steps are described in detail below:

3.3.1 Hypersphere surface sampling

This step is similar to previous approaches in that we first want to find at-least a single failure sample from the failure region. While the previous approaches start with iterative step of small increase in the radii of the annular regions or hyper-spheres to find the first failure samples close to the MPFP. Otherwise, subsequent step of DHS or the consecutive mean shift will require more number of iterations to reach MPFP. Our approach differs here in that we are not interested in finding a failure sample close to MPFP in this stage and so do not sample within hyper-spheres or annular regions. As such, random samples from the *surface* of a hypersphere with a high radius R_0 ($= 5\sigma$) are simulated to find failure samples. If there is no failure sample among the samples then the radius of hypersphere is incremented by 1σ. The objective of this step is to reach a failure sample quickly. Since the sampling is done on the surface of hypersphere, all failure samples are at the same radial distance from the origin and have the same failure-probability (assuming variation in threshold voltages is a normal distribution). So a random sample from these failure samples is selected as our first failure sample for the next step of radial simulation.

3.3.2 Radial simulation

Since the MPFP lies on the failure boundary, the random sampling within a hypersphere centered at the last found failure sample (which is likely far away from the failure boundary) will only waste SPICE simulations. So, to reach the failure boundary, samples are simulated *radially inwards* from the first failure sample till a non-failure sample is reached at which sample we have reached near the pass/fail boundary (between the non-failure sample and the last radially simulated failure sample). The last radially simulated failure sample is then used for spherical surface sampling in the next step.

3.3.3 Spherical-Surface Sampling

Once at the pass/fail boundary, there is no point in doing further radial simulations in that direction. Instead the simulation should be shifted to a different sample at the same radius that has higher probability of being close to MPFP. A surface-spherical random sampling is thus performed on a hypersphere with the radius as the norm of the current failure boundary sample and center as origin. The objective of this random sampling is *NOT* to choose the minimum-norm sample among the failure samples like in previous proposals (in fact all random samples have the same norm because they are at the same radial distance from origin). Rather, it is to find an approximate range of failure region in each angular dimension at the current radius. Then, from the largest failure range found, the failure sample near the middle is chosen for next iteration of radial simulation. Initially, it is not required to find the failure-ranges with high accuracy and so random sampling can be done with fewer number of samples. As we move closer to the origin, the surface area of the hypersphere and the failure-range, both decrease. Thereby increasing the sample density on the surface of the hypersphere. So, the accuracy of range estimation increases as the simulated samples moves closer to the origin.

The failure samples from the spherical sampling are first divided into two sets based on their θ_{n-1} angular coordinate value i.e. for a failure sample whether its $\theta_{n-1} \in [0, \pi]$ or $\in [\pi, 2\pi]$. Then, for each of the rest angular coordinates θ_i, the failure samples in the two sets are sorted by their θ_i coordinate values and stored as the failure range for that angular coordinate in each set, $FailRange[\theta_i|\theta_{n-1} \in [0, \pi]]$ and $FailRange[\theta_i|\theta_{n-1} \in [\pi, 2\pi]]$. The maximum failure-range for each of the angular coordinates $MaxFail[\theta_i]$ is defined as the longest of these two failure ranges,

$$MaxFail[\theta_i] = max(FailRange[\theta_i|\theta_{n-1} \in [0, \pi]], FailRange[\theta_i|\theta_{n-1} \in [\pi, 2\pi]])$$

Furthermore, each $MaxFail[\theta_i]$ again consists of two failure-range lists for $\theta_i \in [0, \frac{\pi}{2}]$ and $\theta_i \in [\frac{\pi}{2}, \pi]$. Some issues arise while finding the failure-range as dimensions of the problem increases:

1. If the failure-ranges for $\theta_i \in [0, \frac{\pi}{2}]$ and $\theta_i \in [\frac{\pi}{2}, \pi]$ of each of the angular coordinates θ_i are estimated then we have to consider that the number of these failure-ranges will grow as an exponential power of 2 with the increasing number of dimensions. To address this problem, we define "Extended Failure Range" for each angular coordinate θ_i

   $$ExtendedFailureRange,\ EF[\theta_i] = [P_L, P_U] \tag{18}$$

   where,

   - P_L is the non-failure sample with the largest θ_i coordinate value which is less than that of all the failure samples. It is the lower bound on failure-range for the angular coordinate θ_i
   - P_U is the non-failure sample with the smallest θ_i coordinate value which is greater than that of all the failure samples. It is the upper bound on the failure-range for angular coordinate θ_i

This $\mathrm{EF}[\theta_i]$ range includes the failure regions as well as the non-failure regions lying in-between for each coordinate θ_i. This trade-off allows for linear scaling of the number of failure-ranges as the number of dimensions increase because of each angular coordinate will have only one range $\mathrm{EF}[\theta_i]$.

2. For each angular coordinate θ_i, we have estimated the $\mathrm{EF}[\theta_i]$ and the subsequent step would be to select a failure sample from one of these extended failure ranges for the next iteration of radial simulation. However, selecting the next sample from $\mathrm{EF}[\theta_i]$ is no longer trivial because it also includes non-failure region. The middle sample from the range could very well be a non-failure sample and so should not be selected as the next sample. To solve this issue, we partition the previously estimated failure-range $\mathrm{MaxFail}[\theta_i]$ of the angular coordinate θ_i with largest $\mathrm{EF}[\theta_i]$ into the two sets for $\theta_i \in [0, \frac{\pi}{2}]$ and $\theta_i \in [\frac{\pi}{2}, \pi]$ and select the largest of the two failure ranges. From this failure-range, we choose the middle sample for radial simulation in the next iteration. Furthermore, in the subsequent random surface-sampling steps, only those random samples that lie in the $\mathrm{EF}[\theta_i]$ of each angular coordinate θ_i are SPICE simulated. This is based on the observation that the length of extended failure-range reduces as the radial distance decreases, which can be seen in Figure 17.

$$\mathrm{EF}[\theta_i|\mathrm{iteration} = j] \subseteq \mathrm{EF}[\theta_i|\mathrm{iteration} = j-1]$$

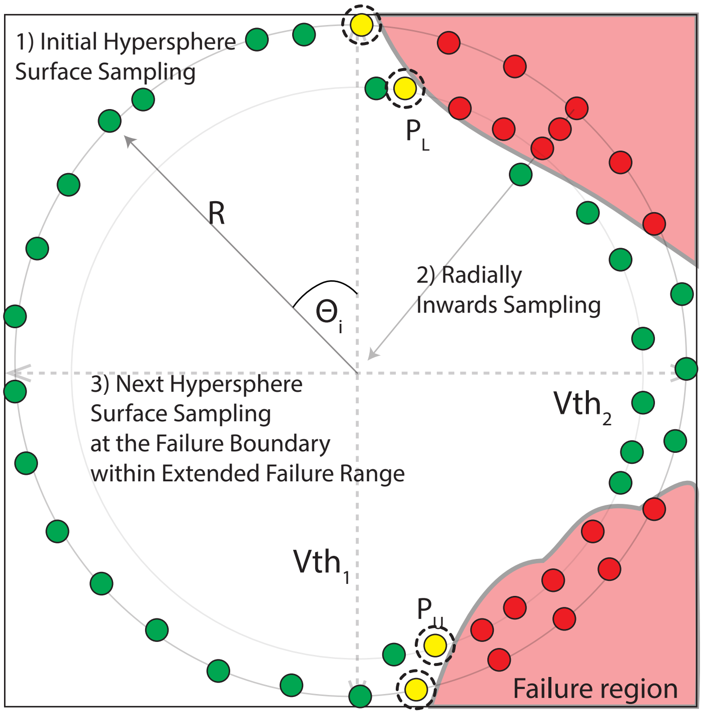

1) Initial Hypersphere Surface Sampling
R
Θ_i
P_L
2) Radially Inwards Sampling
3) Next Hypersphere Surface Sampling at the Failure Boundary within Extended Failure Range
Vth_2
Vth_1
P_U
Failure region

Figure 17: Illustration of SSFB for two-dimensional case.

3.3.4 Termination

When the radially simulated sample reaches the pass/fail boundary and the subsequent spherical sampling results in no failures, then it is an indication that the sample 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 sample crosses the pass/fail boundary. Random sampling is performed again at this new found fail boundary sample and if none of the samples 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 sample found is near the MPFP, then the random spherical sampling is needed only three times with an overhead of 8 simulations before each sampling (Radial step width progresses as: 1σ (no failures) $- >$ go back to last failure sample, do random sampling and take 8 radial steps of 0.125σ to reach non-failure region $- >$ go back to the last failure sample, do random sampling and take 8 radial steps of 0.016σ to reach non-failure region $- >$ finally repeat the same with 2 radial steps of 0.01σ). 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.

```
$StepSize := 1$
$R := R_0$ # 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
    $R$ ←-- $R + 1$
    Uniform Surface Sampling on $EF_\theta$
end while
P :=Random Chosen Fail sample
while $StepSize \not< 0.01$ do
    while $P \neq Pass$ do
        Radially inward simulation from P by $StepSize$
    end while
    $R := Radius(P) + StepSize$
    Uniform Surface Sampling on $EF_\theta$ at radius R
    Find Fail samples
    Separate Fail samples for two Hemispheres $(HEM1, HEM2)$
    ($\theta_{n-1}$:[0,$\pi$],[$\pi$,2$\pi$])
    Foreach $\theta_i$ do
        Fail[$\theta_i$] :=Sorted Fail samples on $\theta_i$
        $P_u$ :=Last Pass before First fail in Fail[$\theta_i$]
        $P_L$ :=First Pass after Last fail in Fail[$\theta_i$]
    end Foreach
    $EF_\theta := [(P_U, P_L)_{\theta_1}...(P_U, P_L)_{\theta_{n-1}}]$
    $\theta_{Max} := \theta_i$ with Max ($[P_U, P_L]$)
    MaxFail[$\theta_{MAX}$]=Max(Fail[$\theta_{MAX}]_{HEM1}$,Fail[$\theta_{MAX}]_{HEM2}$ )
    P :=Middle sample in MaxFail[$\theta_{MAX}$]
    IF $EF_\theta \equiv None$ do
        $StepSize := StepSize/8$
        P :=Last Fail sample
    end IF
$MPFP := P$
```

Algorithm 1: SSFB

3.3.5 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 18.

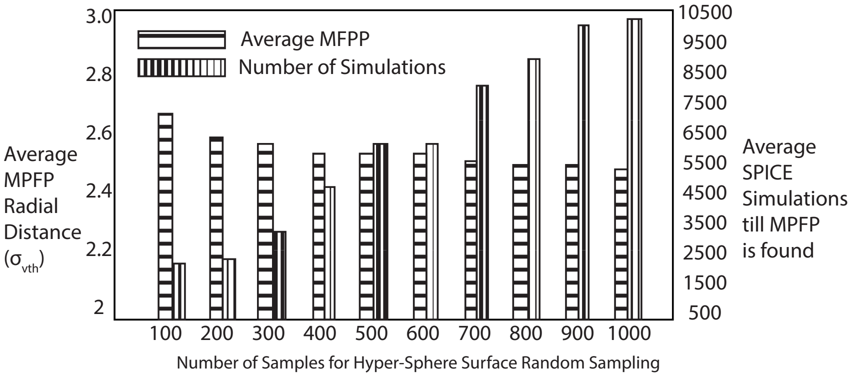


Figure 18: Mean Radial Distance of MPFP vs Total simulations for different sampling options. The radial distance is normalized to σ_{Vth}

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

3.3.6 Read-Failure Probability

The read-failure probability estimate using the Importance Sampling at the MPFP found by SSFB method is compared with the methods Incremental-Hypersphere-Sampling (IHS) Decremental-Hypersphere-Sampling(DHS), Consecutive Mean-Shift method and particle filter method (Sequential Importance Sampling) is Figure 19. The read-failure analysis of the six-transistor SRAM bitcell operating at 0.6V is repeated 20 times to compare the variance of the MPFP estimate. In the case of IHS-DHS method, 10^4 samples are simulated for each IHS and DHS based on [Hagiwara et al., 2010]. For the consecutive mean-shift method, $2.5 * 10^4$ simulations are performed during the first hypersphere sampling and for rest of the steps 10^4 simulations are performed, [Kida et al., 2012] . For particle filter method, sampling-resampling algorithm is run with 500 particles, [Katayama et al., 2010]. The results in Figure 19, show that the SSFB method has a similar variance as the Mean-Shift method and a smaller variance than the Seq-IS method, moreover, reduces the SPICE simulation cost by 40x. Since, SSFB always terminates exactly at the pass/fail boundary, the estimate of the MPFP by the SSFB is the lowest of all.

Table 1: Average number of simulations for finding MPFP

MPFP estimation Method	Average #Simulations	Runtime
Proposal	2078	1m33s
IHS-DHS	$8.3 * 10^4$	7m37s
Mean-Shift	$7.7 * 10^4$	6m11s
Seq-IS	4728	3m57s
Runtime on 4-thread 2-core 2.5GHz processor, L1-32KB, L2-256KB		

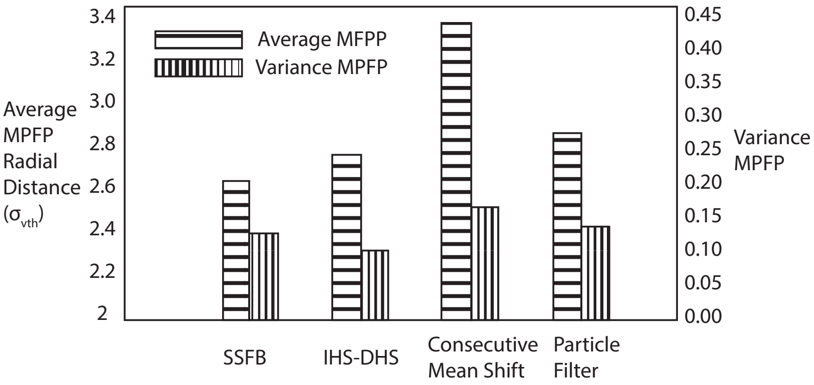


Figure 19: Accuracy Comparison of Proposal with IHS-DHS and mean-shift method. MPFP_{AVG} is the mean radial distance of MPFP and MPFP_{VAR} is the variance of radial distance of MPFP for 20 different runs.

3.4 REEM (Region Estimation by Exploiting Monotonicity)

The objective of this method is to estimate the fail-area and non-fail area before the Monte-Carlo simulations. Once the area is estimated then Importance Sampling methods can be used to estimate failure probability without having to do actual spice simulations. For every sample, we can determine that it is a failure sample if it lies inside the failure region (“sample" refers to a point in the parameter sample space).

REEM is different from other boundary estimation methods because it exploits the *Monotonicity* [Khalil et al., 2008] property of SRAM failure to estimate the regions. A consequence of this property is that, given a simulated failure sample, we can determine a part the failure region in the threshold voltage variation space i.e. region containing all the samples with larger values of variations than the given failure sample. Similarly for any non-failure sample we can determine a part of the non-failure region consisting of samples with lesser values of threshold voltage variation. Thus with every sample in the parameter space we have an estimate of the failure/non-failure region covered by that sample. The failure and non-failure regions can thus be estimated by choosing samples sequentially from the parameter space.

3.4.1 Metric to choose among samples

With sequential selection of samples the parameter region is divided into three parts:

- *Fail Region*
- *Non Fail Region*
- *Unknown Region*

To find the next sample for spice simulation, random sampling is done within the unknown region. From these random samples, one sample is picked which can give the largest possible increase in failure or non-failure region region.

3.4.1.1 REEM: Monte-Carlo Approach

To determine the next sample location for SPICE simulation, the metric *"Contribution"* is introduced.
For sample P on parameter space:

$$\begin{aligned} \mathrm{Contribution}(P) = [\Delta \mathrm{Area}_{fail}(P) * \mathrm{Prob}_{fail}(P) * \\ \prod_{i\_dim}^{NDIM} \mathrm{CDF}(|P[i\_dim]|)] * \\ [\Delta \mathrm{Area}_{nonfail}(P) * \mathrm{Prob}_{nonfail}(P) \\ * \prod_{i\_dim}^{NDIM} (1 - \mathrm{CDF}(|P[i\_dim]|)] \end{aligned} \tag{19}$$

The first part of the equation for *Contribution* corresponds to the "contribution" by the new sample to the failure region. The second part corresponds to the "contribution" to the non-failure region. The product of both is taken so that the method does not get stuck in increasing only one of either fail or non-fail areas.
Among the samples, the one which maximizes the *Contribution* is selected. After running the simulation of the chosen sample, depending on whether the sample was fail/non-fail the estimates of the three regions are updated. This goes on till the unknown region goes below a certain percentage to obtain desired level of accuracy.
The parts of the metric are described below:

$\Delta Area$ The test samples on the parameter space which provide larger increase in current Failure/Non-Failure area should be preferred over the rest. With this policy, most of the parameter space can be covered with fewer simulation samples. The problem, then, is of estimating the increase in area coverage for each of the test samples.

- Estimating $\Delta Area_{fail}$: First the failure samples inside the fail area covered by the test sample, p_n, are determined. All these *covered fail samples* will be replaced by the test sample if the test sample is later found by simulation to be a fail sample. Thus, in essence, an outer cover of the fail region is maintained. For the remaining *effective fail samples*, we determine the $\Delta Area_{fail}$ by equation 20.
The, $\Delta Area_{fail}$ is estimated for the sample p_n, when we have the *effective fail samples* $\{p_1 \ldots p_{n-1}\}$. For every iteration of the outer sum on number of dimensions, $NDIM$, the samples $\{p_1 \ldots p_n\}$ are sorted on that dimension in *decreasing order*.

$$\begin{aligned} \Delta Area_{fail}\{p_n | p_1 \ldots p_{n-1}\} = Area_{fail}\{p_1 \ldots p_{n-1}\} - \\ \sum_{i\_dim}^{NDIM} \sum_{p_i}^{Sorted\_p_n} I_{i\_dim}(p_i) * Fail\_Boundary\_Area(p_i, i\_dim) \end{aligned} \tag{20}$$

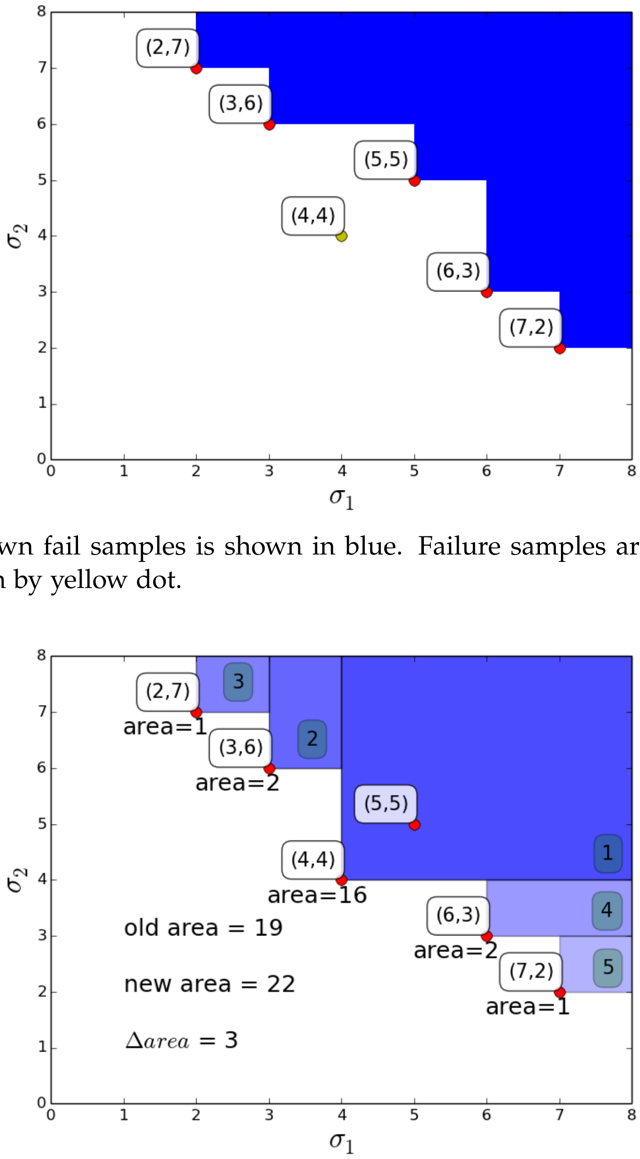
$where$

$$I_{i\_dim}(p) = \begin{cases} 1 & if \quad p_i[i\_dim] \leqslant p_n[i\_dim] \text{ and} \\ & p_i \text{ is not covered in past iterations} \\ & \text{of summation over sorted\_}p_n \text{ in eq 20} \\ 0 & otherwise \end{cases}$$

and,

$$Fail\_Boundary\_Area(p_i, i\_dim) = (p_{i-1}[i\_dim] - p_i[i\_dim]) * \prod_{j\_dim \neq i\_dim}^{NDIM} (\sigma_{max}[j\_dim] - p_i[j\_dim]) \tag{21}$$

An illustration of estimating $\Delta Area_{fail}$ is shown in figure 20



(a) Area covered by known fail samples is shown in blue. Failure samples are shown by red dots. Test sample $\{4,4\}$ is shown by yellow dot.

(b) $\Delta Area_{fail}$ estimated for the test sample $\{4,4\}$ according to eq 20. Fail sample $\{5,5\}$ is *covered* by the test sample. The iterations of the algorithm are shown in the green boxes which show how the fail region is partitioned.

Figure 20: $\Delta Area_{fail}$ illustration for a 2D example

- Estimating $\Delta Area_{nonfail}$: Estimation of $\Delta Area_{nonfail}$ is similarly done with equation 22 for the test sample p_n where the *effective non-fail samples are* $\{p_1 \dots p_{n-1}\}$. The sorting of samples for the outer sum on NDIM is done in *increasing order* in that dimension.

$$\Delta Area_{nonfail}\{p_n|p_1 \dots p_{n-1}\} = Area_{lnonfail}\{p_1 \dots p_{n-1}\} - \sum_{i\_dim}^{NDIM} \sum_{p_i}^{Sorted\_p_n} I_{i\_dim}(p_i) * NonFail\_Boundary\_Area(p_i, i\_dim) \tag{22}$$

where,

$$I_{i\_dim}(p) = \begin{cases} 1 & \text{if } \; p_i[i\_dim] \geqslant p_n[i\_dim] \text{ and} \\ & p_i \text{ is not covered in past iterations} \\ & \text{of summation over sorted\_}p_n \text{ in eq 22} \\ \\ 0 & \text{otherwise} \end{cases}$$

and,

$$\text{NonFail\_Boundary\_Area}(p_i, i\_dim) = (p_i[i\_dim] - p_{i-1}[i\_dim]) * \prod_{j\_dim \neq i\_dim}^{NDIM} (p_i[j\_dim]) \quad (23)$$

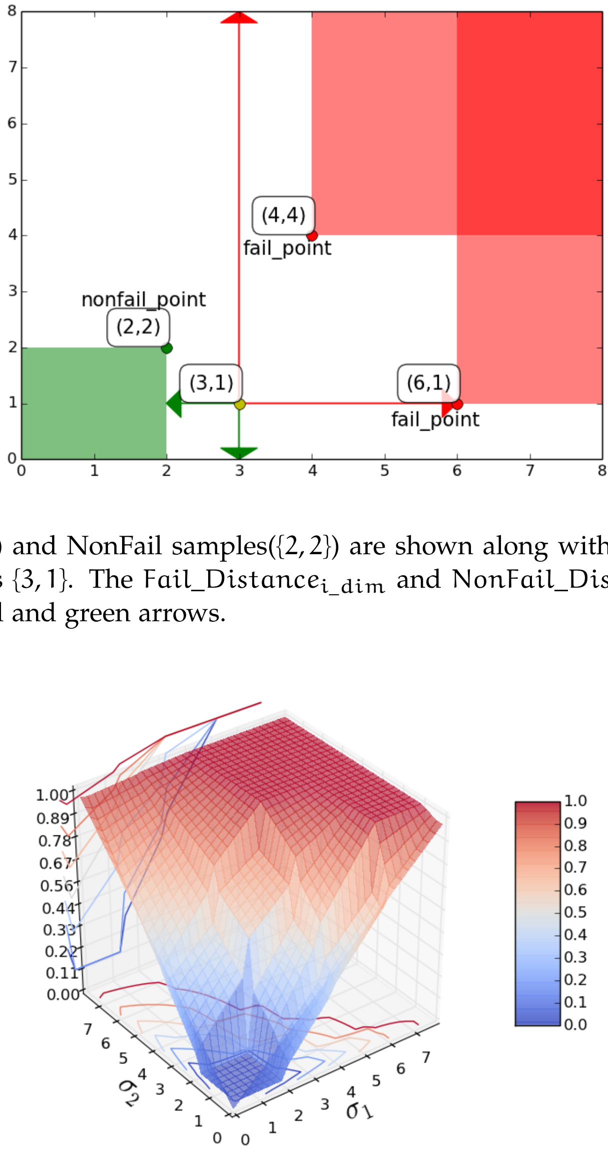
FAIL PROBABILITY OF A SAMPLE When choosing samples for simulation, to increase the fail area, a choice has to be made among test samples which provide same increase to the fail area. In this case, the test sample which is most *likely* to fail must be selected for simulation. Similarly for non fail area, sample which is most *likely* to be *not* a fail sample must be selected among the samples which provide same increase to non-fail area.
For every sample in the *unknown region* of the parameter space, we can approximate how *likely* that sample is a fail sample by measuring the *closeness* of that sample to the fail region and non-fail region. Samples closer to the fail region are more likely to fail, while samples closer to non-fail region are less likely to fail. The REEM method measures the likelihood of a test sample to be a fail sample by equation 24.

$$\text{Prob}_{non-fail}(p_n|\{p_1 \dots p_{n-1}\}) = \prod_{i\_dim}^{NDIM} \frac{\text{Fail\_Distance}_{i\_dim}(p_n)}{\text{Fail\_Distance}_{i\_dim}(p_n) + \text{NonFail\_Distance}_{i\_dim}(p_n)}$$

and,

$$\text{Prob}_{fail}(p_n) = (1 - \text{Prob}_{non-fail}(p_n)) \quad (24)$$

where,
$\text{Fail\_Distance}_{i\_dim}$ is the shortest distance from the test sample p_n to the failure boundary in dimension i_dim. This is shown in figure 21a, where $\{3, 1\}$ is the test sample. The shortest distance from the test sample to Fail, NonFail regions is shown by the red and green arrows respectively for both dimensions in the figure.



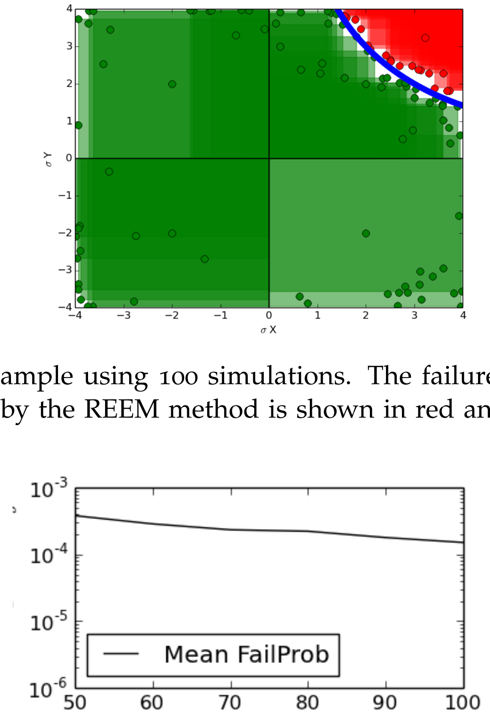
(a) Fail samples($\{4,4\},\{6,1\}$) and NonFail samples($\{2,2\}$) are shown along with the area covered by these samples. Test sample is $\{3,1\}$. The $\mathtt{Fail\_Distance}_{\mathtt{i\_dim}}$ and $\mathtt{NonFail\_Distance}_{\mathtt{i\_dim}}$ are shown for the test sample with red and green arrows.

(b) Probability for samples on the parameter space to be a fail sample is shown. samples near the non-fail sample $\{2,2\}$ have lower probability of being a fail sample. Similarly, samples near the fail samples ($\{4,4\},\{6,1\}$) have higher probability of being a fail sample.

Figure 21: Illustration of failure probability of samples for a 2D example.

CDF OF A SAMPLE Finally, a choice has to be made among samples which are equally probable to fail and also which provide similar increase in the failure/non-failure region.
For these samples, the one with higher probability of being sampled under the Monte-Carlo simulation should be preferred. Choosing samples which have higher probability under the parameter distribution means that the area covered by these samples will be part of region from where majority of samples will be sampled by Monte-Carlo method.

STARTING SAMPLE When choosing the starting sample, the objective is to cover as large area as possible with this starting sample. The starting sample can be randomly sampled from the parameter space. However then the starting area will also be random, meaning that we could possible start with a very small estimate of area.
An alternative method would be to choose the sample $\{\frac{\sigma_1}{2} \ldots \frac{\sigma_n}{2}\}$ which covers equal area in either case if its fail sample or non fail sample. And so, this sample can be taken as the starting sample in the method. Note that, for a N-Dimensional problem there will be 2^N starting samples, one for each quadrant.
The results of the REEM Monte Carlo method for a simple 2D example are shown in figure 22. Once the areas are estimated, we compute the failure probability by using the Monte Carlo method with 10^5 samples. These Monte Carlo samples, however do not need to be simulated as we already know whether they will fail or non-fail depending on which area they fall into. Also, for the samples lying inside unknown region, we know the probability of them being a fail sample (eq 24), which is used to decide whether to take them as fail sample or non-fail sample.



(a) Estimated area for a 2D example using 100 simulations. The failure boundary is shown by the blue line. Fail region estimated by the REEM method is shown in red and the non-fail region is shown in green.

(b) Convergence of the Failure probability estimated as the simulations are increased from 50 to 100. The Monte Carlo failure probability for the example is $1.6 * 10^{-4}$, and the failure probability estimated by REEM with 100 simulations is $1.5 * 10^{-4}$

Figure 22: Results obtained from REEM on a simple 2D example.

LIMITATION OF MONTE CARLO APPROACH The proportion of area that is covered by a sample decreases as the dimensions (variable) in the analysis increase. For example, in the 2-D case the area covered by the middle sample of a quadrant is 25% in either case if the sample is fail or non-fail. However, for 6-D case, this percentage drops to less than 2% of the total area. Thus estimating all the fail/non-fail region is not practical for higher dimensions. As an example with about 1000 spice simulations, about 94.34% of the total area remains unknown for 6D case (i.e. 6T bitcell analysis).

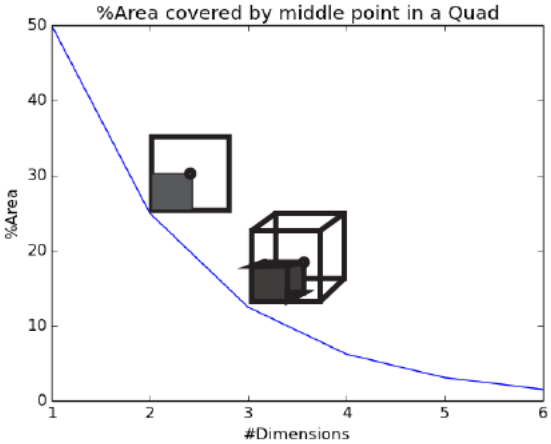


Figure 23: Reduction in the proportion of area covered by the middle sample in the quad $(\{\frac{\sigma_1}{2} \dots \frac{\sigma_n}{2}\})$. The percentage of area covered decreases from 25% for 2D case to just 1.5% in 6D case.

3.4.1.2 REEM: MPFP+IS: Improvement over Monte Carlo approach

As discussed in the last section, the Monte Carlo approach of estimating the entire fail/non-fail region does not scale with increasing dimensions as the proportion of area covered by samples decreases with increasing dimensions.

However, if instead of Monte Carlo we use Mean-shift Importance sampling based methods, we only need to estimate the region of space close to the shifted mean location. Thus the area to be estimated can be restricted to be within a certain region from where most samples will be sampled for Importance Sampling .

For, MPFP based Mean-shift Importance-sampling methods, the target sample is the minimum norm sample in the failure boundary. Thus, only the region near that minimum norm sample has to be estimated. Regions far from MPFP will not provide much samples and thus can be ignored.

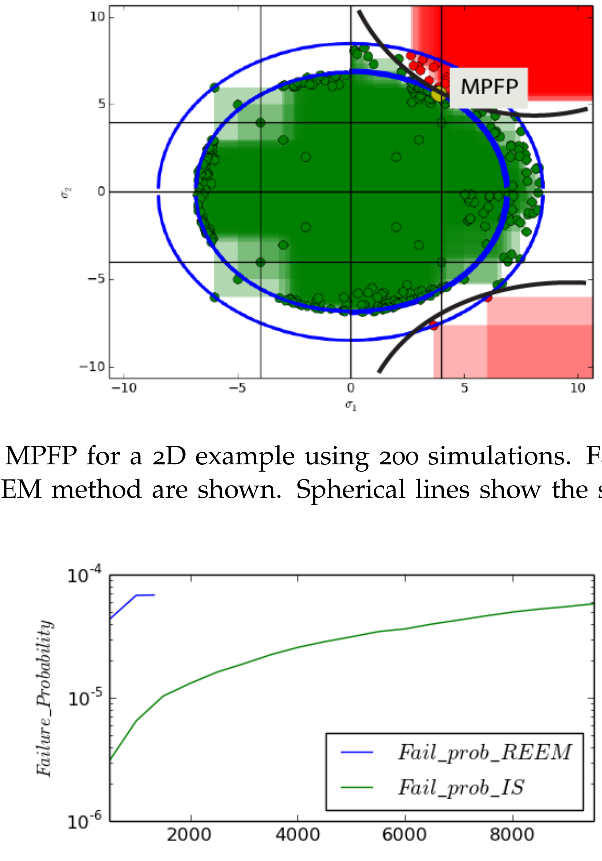
The REEM, thus, starts with the simulation of the samples in the middle of each quadrant. If none of the starting samples fail then we take a step in diagonal direction in each quadrant till atleast one sample of all the starting samples fail. Clearly, the MPFP (which is the minimum-norm sample in the parameter space) will lie within the radial distance of the found failure sample.

Once the failure samples are found among the starting samples, the next step is to converge to the MPFP sample within 0.01σ accuracy. For this, REEM finds the minimum norm sample among the failure samples and then sequentially samples only within the radial distance of this minimum norm sample. Simultaneously, with every HSPICE simulation of a sample, we keep updating the estimates of Fail/Non-Fail region and the radial distance within which random

sampling is to be done.
The contribution function is updated as follows to find the MPFP sample:

$$\mathtt{Contribution}(P) = \Delta\mathtt{Area}_{\mathtt{nonfail}}(P) * \mathtt{Prob}_{\mathtt{fail}}(P) * \prod_{\mathtt{i\_dim}}^{\mathtt{NDIM}} (1 - \mathrm{CDF}(|\mathrm{P}[\mathrm{i\_dim}]|)) \tag{25}$$

The rest of the part of updating the failure/Non-failure area and estimating the failure probability of a single sample remains the same as the Monte Carlo approach.
Once the MPFP sample is located, Importance Sampling(IS) is performed at MPFP location. The N samples for IS are first sampled from the shifted distribution. These N samples are then simulated iteratively in group of 500 samples. samples in a group are selected based on the original contribution function defined in equation 19. With the simulation of every group, fail/non-fail area estimates are updated. Illustration of the method for 2D case is shown in figure 24a.



(a) Estimated area near MPFP for a 2D example using 200 simulations. Fail region and Non-Fail region estimated by the REEM method are shown. Spherical lines show the surface of N-ball within which sampling is done.

(b) For the 2D illustration above, REEM method converges to a failure probability estimate of $6.82 * 10^{-05}$ in 1534 simulations. The Traditional Mean-Shift-Importance-Sampling(i.e. MPFP+IS) converges to failure probability estimate of $5.78 * 10^{-05}$ in 10^4 simulations.

Figure 24: REEM + Importance Sampling

	Simulations			
	MPFP	IS ($4 * 10^4$)	Total	Time
Read Probfail				
REEM P_{fail}=2.4 $*$ 10^{-6}	624	9118	9742	9.2 min
	2 min	7.2 min		
Mean-Shift IS P_{fail}=4.5 $*$ 10^{-6}	$6 * 10^4$	$4 * 10^4$	10^5	17.2 min
	10.4 min	6.8 min		
Write Probfail				
REEM P_{fail}=7.63 $*$ 10^{-5}	545	13453	13998	10 min
	1.6 min	8.3 min		
Mean-Shift IS P_{fail}=8.2 $*$ 10^{-5}	$6 * 10^4$	$4 * 10^4$	10^5	17 min
	10 min	6 min		

Table 2: Simulation comparison of REEM with mean-shift IS method for 6T bitcell. Importance-Sampling stage is done with $4 * 10^4$ samples for both methods. Of these samples, REEM only needed to simulate 9118 samples for Read analysis. Below the simulation count for each stage(MPFP/IS), the run-time for that stage is also mentioned.

3.4.2 Evaluation for 6T SRAM

3.4.2.1 Methodology

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 process in HSPICE circuit simulator. The static read noise margins analysis of the bit cell are computed through the N-Curve method.

3.4.3 Results

The read and write failure probability are calculated using the REEM method and they are compared with the Importance Sampling based method. We simulate a 6T bitcell operating at 0.6V. For the Mean-Shift Importance Sampling based method, roughly 60,000 simulations are needed to estimate the MPFP sample within 0.01σ accuracy [Kida et al., 2012]. Then, IS sampling has to be performed with mean at this estimated MPFP which is done with 40,000 simulations, totaling 10^5 simulations overall.

The REEM method only requires 624 simulations to find the MPFP (Table 2). The importance sampling is done here also with 40,000 simulations, however with REEM, some samples are already known to be fail/non-fail samples before-hand as they fall within the estimated fail/non-fail region. Thus out of the total 40,000 simulations only the 9118 simulations are needed (samples which fall in the unknown region). Thus REEM method provides overall reduction of about 10x in simulation count.

3.5 Conclusion

In this chapter, we introduced SSFB, a sampling method to find MPFP faster; and REEM, a method for estimating the Fail/Non-Fail region in the parameter space. SSFB reduces SPICE simulations by sampling only near failure boundary and uses extended failure ranges to find MPFP. REEM finds the MPFP sample and also estimates the fail/nonfail region around that sample. That is followed by an Importance Sampling step around the MPFP. Samples which fall inside the estimated fail/nonfail region during the Importance Sampling step do not need to be simulated with Spice as they are already known to be fail/nonfail. To evaluate the effectiveness of REEM , its SRAM read/write failure analysis results were compared with Mean-shifted Importance Sampling based methods and an overall reduction of 10x in simulation count was achieved.

Essentially, all models are wrong, but some are useful.

George E. P. Box

4 Surrogate modeling of SRAM memory margins using Gaussian process regression

4.1 Introduction

In this chapter we introduce and motivate the statistical modeling of memory noise margins that will be used to create surrogate models for estimating memory failure probability. Probabilistic inference is a method for learning a function from data which takes a hypothesis set and compares how well the input data is fit by the models in the hypothesis set. This comparison can be used to extract structure such as additive, symmetry, or periodicity present in the high-dimensional input data.

At the ultra-low voltages, the SRAM bit-cell read current has an exponential dependence on the threshold voltage[Calhoun et al., 2005]. Thus, the presence of threshold voltage variations results in non-linear response of dynamic margins to these variations. Gaussian process [Rasmussen, 2006] can be used to make very flexible models using universal kernels, such as the six-dimensional Radial Basis function kernel (RBF) which can be used to model any six-dimensional continuous function [Micchelli et al., 2006]. However with increasing dimensions, Gaussian process regression suffers from the curse of dimensionality [Geenens et al., 2011], that is, the required number of training samples needed to correctly model the function increase exponentially and thus the regression becomes slower. In this chapter we present a methodology to build surrogate models of the non-linear behavior of SRAM dynamic noise margins at sub-threshold voltages using additive kernel based Gaussian Process regression [Duvenaud et al., 2011]. An additive kernel is a positive-definite function that decomposes into the sum of low-dimensional kernels. This surrogate model can then be used to estimate memory margin failure probabilities using either traditional Monte-Carlo or Importance Sampling techniques [Kanj et al., 2006, Dolecek et al., 2008].

The chapter is organized as follows. In section 4.2 related work is discussed and the relevant background material is presented in section 4.3. In section 4.4, the proposed method is discussed as three-dimensional and six-dimensional case study for modeling dynamic read margin of 6T SRAM cell. Probability failure results are given in section 4.5. And finally, conclusions are given in the section 4.6.

4.2 Related Work

Traditionally, reduction in the needed spice simulations compared to the Monte-Carlo method is achieved by improving the sampling process such as by using Importance Sampling based methods (Mixture Importance Sampling [Kanj et al., 2006], Minimum-Norm Importance Sampling [Dolecek et al., 2008]) or extreme value statistics [Singhee and Rutenbar, 2008] to estimate the margin failure probabilities of the SRAM cells. However these methods still need tens of thousands of spice simulations to estimate very low failure probabilities ($< 10^{-6}$) of SRAM noise margins. The effectiveness of using Kriging (a spatial regression technique similar to Gaussian Process regression) in reducing the spice simulations by building highly accurate meta-models was shown in [Okobiah et al., 2014]. The work, however, used simple kriging to build meta-models and a simple spherical covariance function was used to build the covariance matrix. Furthermore, Importance Sampling from surrogate models have also been proposed for faster high-sigma yield analysis of the SRAM cells, such as [Yao et al., 2013] which uses Radial Basis Function (RBF) kernel network to build the surrogate model. Our method focuses on improving this modeling process by finding the optimum kernels (covariance functions) to build the surrogate models. Our method provides an alternative to the universal kernels such as RBF used in these previous approaches. Higher model accuracy with smaller out-of-sample error than the RBF kernel (that is, better extrapolation capability away from the sampled data) is achieved by tuning the covariance kernel of the Gaussian process to the SRAM margins.

4.3 Background

4.3.1 Gaussian Process Regression

A Gaussian Process (GP) used for non-parametric regression [Rasmussen, 2006] is a collection of random variables (the value of the function $f(x)$ at location x) where each random variable has Gaussian distribution and any finite set of random variables $f(x_1), f(x_2), f(x_3), ..f(x_n)$ have a joint Gaussian distribution.

A GP is specified by a mean function,

$$E[f(x)] = \mu(x) \tag{26}$$

and a covariance function,

$$COV(f(x_1), f(x_2)) = k(x_1, x_2) \tag{27}$$

where the function $k(x_1, x_2)$ is called the kernel. The kernel determines the complexity of the distribution over functions generated by the GP after conditioning on the input data. This complexity over function distribution is also referred as the capacity of the GP regression. A model with high capacity is better than a model with low capacity when modeling highly non-linear functions.

The model selection (comparison between models generated by a GP) is done by calculating the marginal likelihood of the input data given a particular model from the hypothesis set. The marginal likelihood of function values [Duvenaud, 2014] $f(x_1), f(x_2), ..f(x_n)$ at location $X = (x_1, x_2, ...x_n)$ with a GP prior of mean function- $\mu(x)$ and covariance function- $k(x_1, x_2)$ is given by ,

$$\begin{aligned} P(f(X)|X, \mu, k) = N(f(X)|\mu(X), k(X, X)) \\ = (2\pi)^{-n/2} * \\ |k(X, X)|^{-1/2} * \\ e^{(-1/2(f(X)-\mu(X))^T k(X,X)^{-1}(f(X)-\mu(X)))} \end{aligned} \tag{28}$$

The prediction distribution [Duvenaud, 2014] of function f at test point $x*$ by the GP posterior of mean function- $\mu(x)$ and covariance function- $k(x_1, x_2)$ conditioned on the input data $f(x_1...x_n)$ at locations $X = (x_1...x_2)$ is given by,

$$\begin{aligned} P(f(x*)|f(X), X, \mu, k) = N(f(x*)| \\ \mu(x*) + k(x*, X)k(X, X)^{-1}(f(X) - \mu(X)), \\ k(x*, x*) - k(x*, X)k(X, X)^{-1}k(X, x*)) \end{aligned} \tag{29}$$

4.3.2 Kernel functions

The model capacity of the distribution over functions, generated by GP is determined by the choice of kernel (prior covariance function) used in the modeling of the input data. A kernel has to be a positive-definite function of any two input locations x, x' and specifies the similarity in the values of a function generated by the GP at x and x'. The specific shape of the co-variance function generated by the kernel is determined by the kernel's hyper-parameters.
Some commonly used kernels and the corresponding prior induced the kernels on the function values are discussed below briefly:

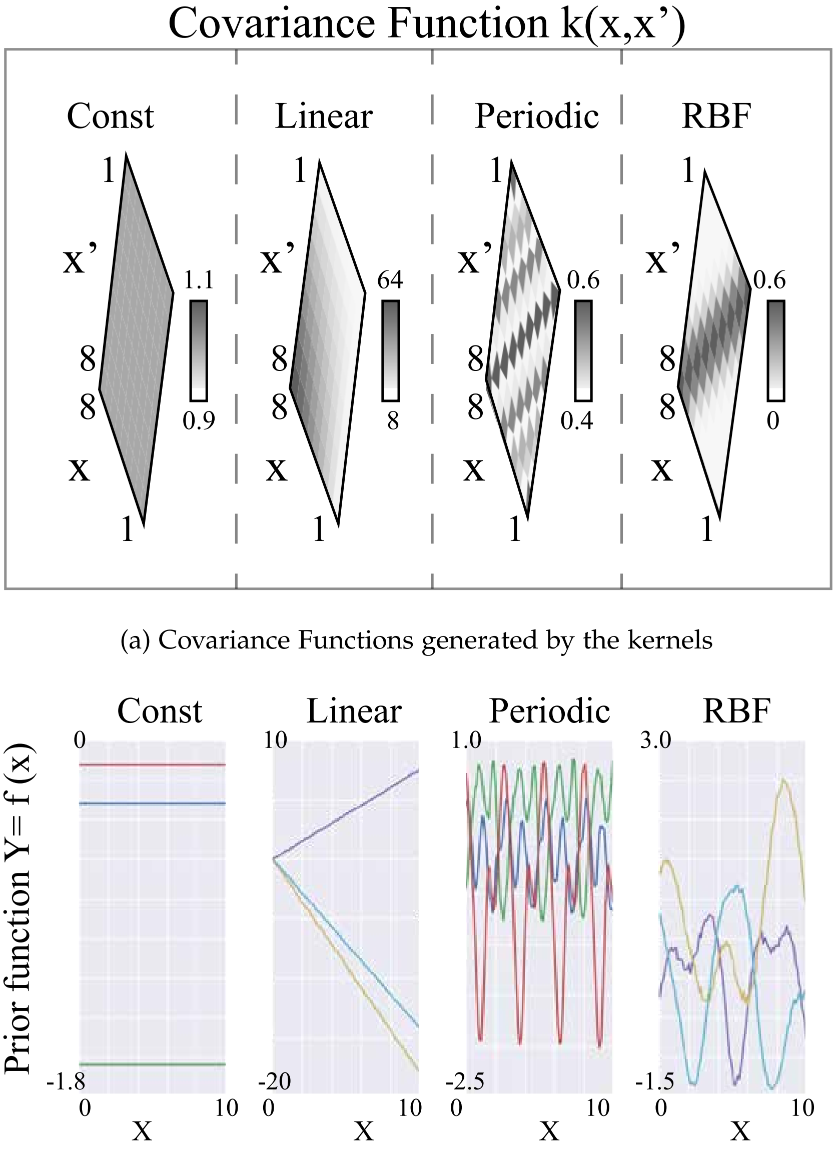
- Constant Kernel: This kernel generates co-variance functions which are constant over the entire input domain, i.e. $k(x, x*) = \sigma$, where the signal variance σ is a constant. GP priors generated with this kernel are constant functions.

- Linear Kernel: This kernel generates linear co-variance functions, $k(x, x*) = \sigma^2 * ||x - x * ||$, with signal variance σ as the hyper-parameter. GP priors with this kernel are linear functions.

- Periodic Kernel: This kernel generates periodic co-variance function,
$k(x, x*) = \sigma^2 e^{((-2sin^2(\pi|x-x*|/p))/L^2)}$, where the hyper-parameters are,
Signal variance, σ, Periodicity (length between repetitions of function), p, and length scale, L, which determines how far the model can extrapolate from the training data.
GP priors with this kernel are periodic functions and can be used to model functions which repeat themselves periodically.

- Radial Basis Function Kernel (RBF)
(or Squared Exponential Kernel
or Gaussian Kernel): This kernel generates exponential quadratic co-variance functions, $k(x, x*) = \sigma^2 exp(-(x - x*)^2/L)$, with hyper-parameters,
Signal variance, σ, and Length scale, L. GP priors are smooth functions having infinitely many derivatives.

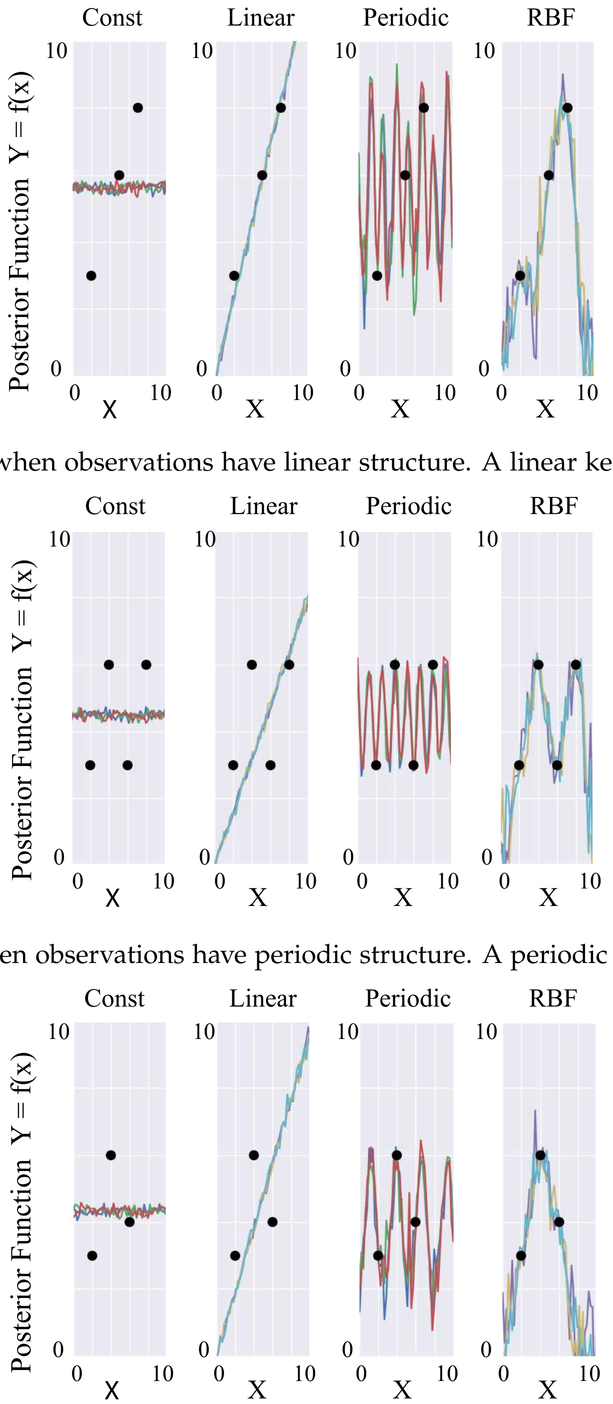
The co-variance functions and a sample of three GP priors induced by these kernels is shown in Figure 25. These kernel functions provide different extrapolation capabilities because of the difference in the structure of their GP prior functions [Duvenaud, 2014]. Figure 26 compares the extrapolation capability of these kernels for input observations with three types of structures - linear, periodic, and non-linear non-periodic structure. It can be seen that for better extrapolation the structure encoded by the kernel should match the trend in the input observations. For example, for periodic observations the repeating structure is preserved in extrapolation only with periodic kernel, While the RBF kernel can interpolate the observed data, it however cannot extrapolate the function values at locations farther from the observed data [Wilson, 2014].



(a) Covariance Functions generated by the kernels

(b) Sample GP priors induced by these covariance functions

Figure 25: The figures show A) the types of co-variance functions, B) and the corresponding GP priors induced by the kernels. A sample of 3 GP priors is shown as representative of the hypothesis set induced by the kernels



(a) Posterior functions when observations have linear structure. A linear kernel captures the trend.

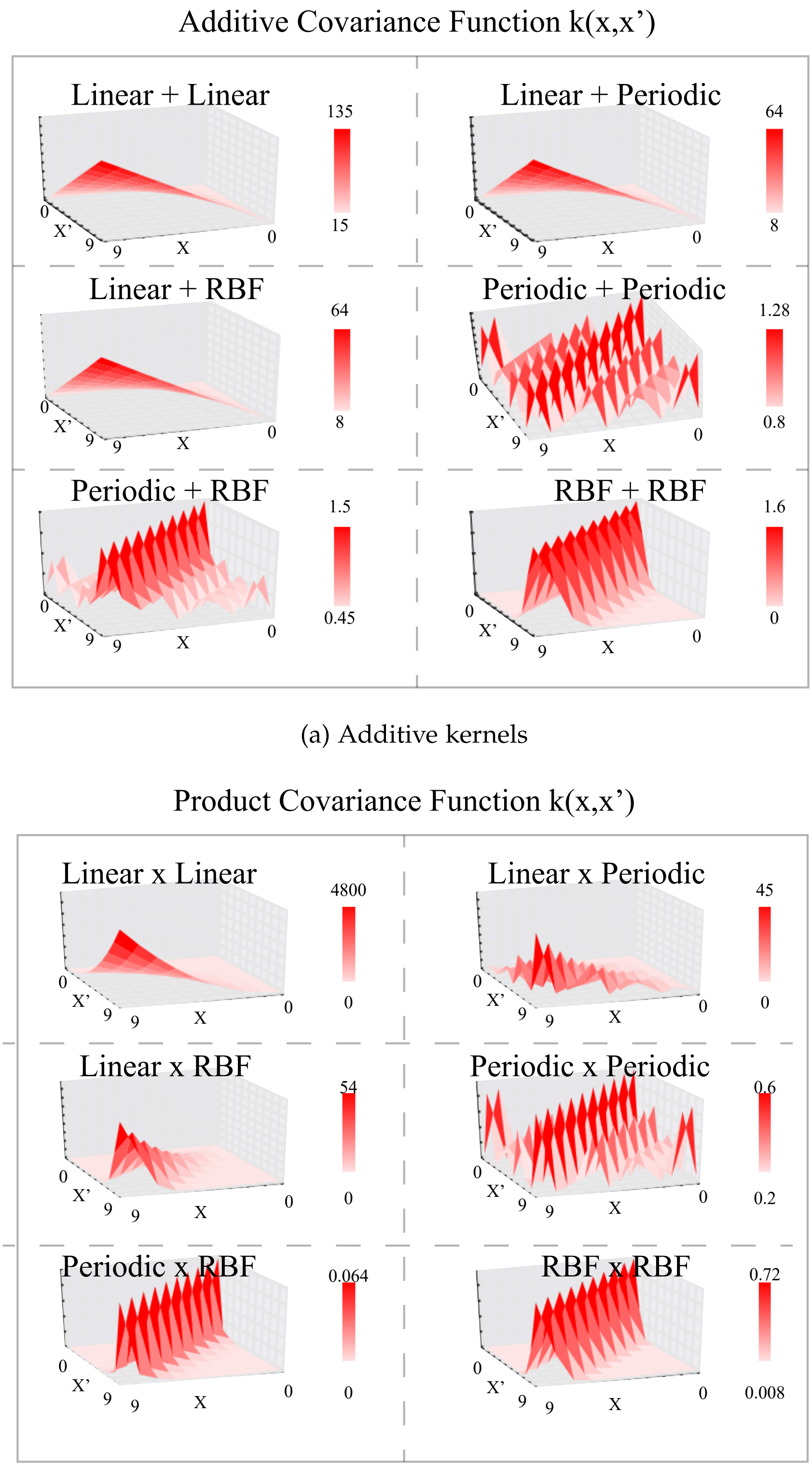
(b) Posterior functions when observations have periodic structure. A periodic kernel captures the trend.

(c) Posterior functions when observations have non-linear and non-periodic structure. None of the linear or periodic kernels capture the trend.

Figure 26: Each sub-figure shows three posterior functions obtained when the three GP priors shown in the Figure 25b are conditioned on the input data observations. Using kernels with similar structure as that present in the data observations enables extrapolation at unknown data locations. All the three sub-figures show that while RBF kernel can interpolate the observed data, it cannot extrapolate at locations outside the observed data range.

4.3.3 Composite Kernels

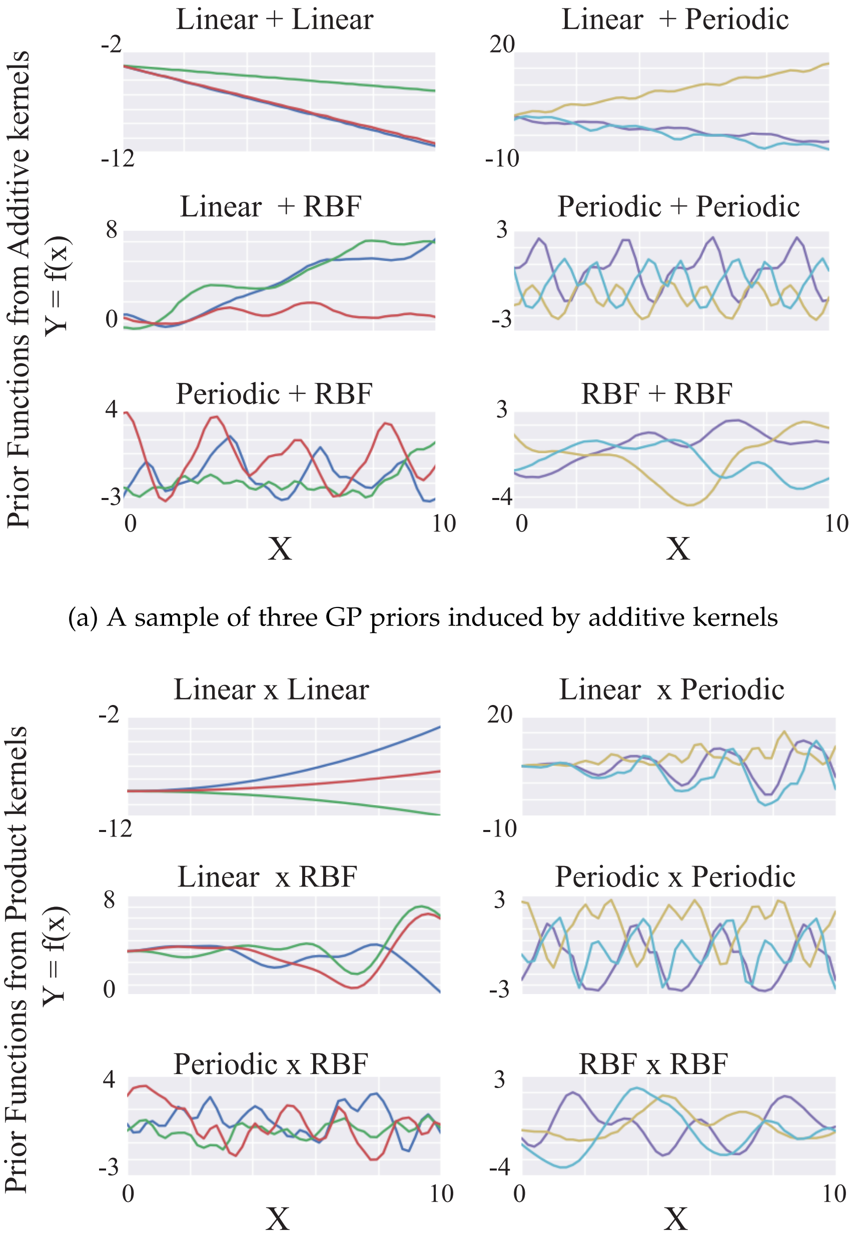
The kernels discussed in the previous section provide prior and posterior functions with simple structures such as constant, linear and periodic functions. RBF kernel provides all smooth infinitely differentiable functions as prior. Only using constant, linear or periodic kernels is not sufficient to model observed data with non-linear structure such as quadratic functions. Composite kernels can be created by adding kernels or by taking the product of the kernels. The co-variance function of the additive kernel takes on the properties of the dominant kernel among the constituent kernels. The maximum covariance function values for Linear (65) > RBF (0.8) > Periodic (0.6) in the input domain [0,10]. As such, in the case of additive kernels, Linear kernel dominates over these kernels e.g. Linear+Periodic, and Linear+RBF in Figure 27a. The GP priors of these two additive kernels will be linear with local periodic structure, and linear with local function variations respectively. This is seen in Figure 28a. The co-variance function generated from the product of the kernels takes high value only at those input locations where all the kernels in the product have high values. Otherwise, its value decreases more than the additive kernel, as seen in Figure 27b. Because of this, the extrapolation capability of the product kernels is lower than that of the additive kernels as is illustrated by the posterior functions of Linear+RBF in Figure 29a vs Linear*RBF in Figure 29b for observed data sampled from a quadratic function. The product kernels give flexible GP priors with larger local changes in function values than the GP priors from additive kernels. Overall, it is seen that the product kernels provide more flexible prior functions, while the additive kernels can extrapolate from the observed data to locations farther from observed range [Duvenaud, 2014]. The sum of product kernels, thus, provides the extrapolation capability of the additive kernels and also the flexible priors from the product kernels.



(a) Additive kernels

(b) Product kernels

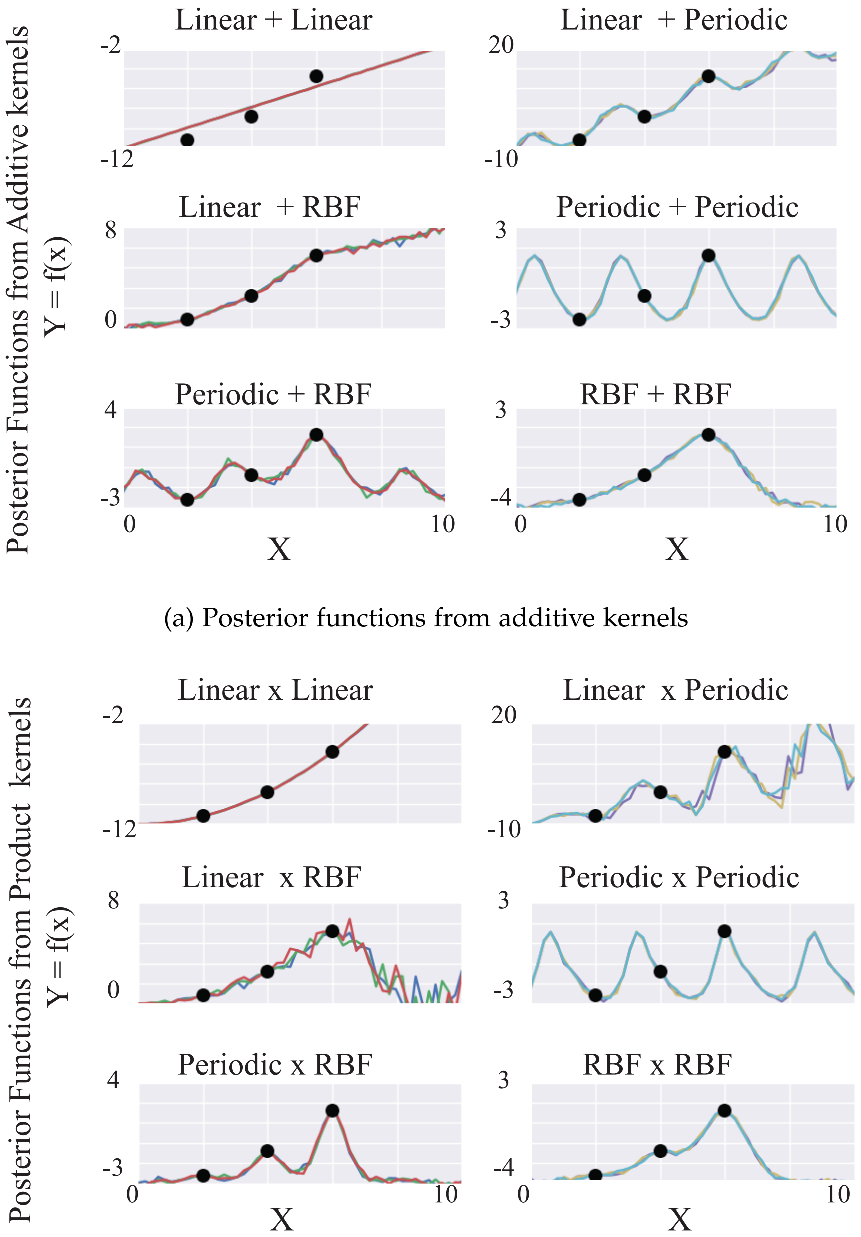
Figure 27: Co-variance functions generated from Additive and Product composite kernels. The co-variance function values for the product kernels decrease faster than the additive kernels, this reduces their extrapolation capability at points outside the observed data range.



(a) A sample of three GP priors induced by additive kernels

(b) A sample of three GP priors induced by product kernels

Figure 28: GP priors provided by the composite kernels. Linear+Linear gives again linear priors. Linear+Periodic gives prior functions with locally periodic and linear trend. Linear+RBF gives prior functions with local smooth changes and linear trend. The Linear*Linear gives quadratic prior functions.



(a) Posterior functions from additive kernels

(b) Posterior functions from product kernels

Figure 29: Three posterior functions for the composite kernels with their GP priors (in Figure 28) conditioned on three input data points sampled from a quadratic function. Linear*Linear captures the quadratic trend. Comparing additive and product kernels, Linear*RBF and Linear+RBF, the additive kernel is capable of extrapolating the increasing trend in the data, while the product kernel is not able to extrapolate outside the observed data range.

4.4 Modeling 6T SRAM dynamic margins at sub-threshold voltage

4.4.1 Three dimensional Case

Three threshold voltage variation sources in 6T bit-cell are considered: pull-up (PMOS), pull-down (NMOS) and the access transistor. Both the pull-up transistors are given the same variation value, and similarly for pull-down transistors and access transistors. The dynamic read margin for this analysis is defined as the voltage difference between the node storing logic value "1" and the node storing logic value "0" at the end of 20ns read word-line pulse width. The sensitivity analysis of the dynamic read margin for the three variation sources is shown in Figure 30.

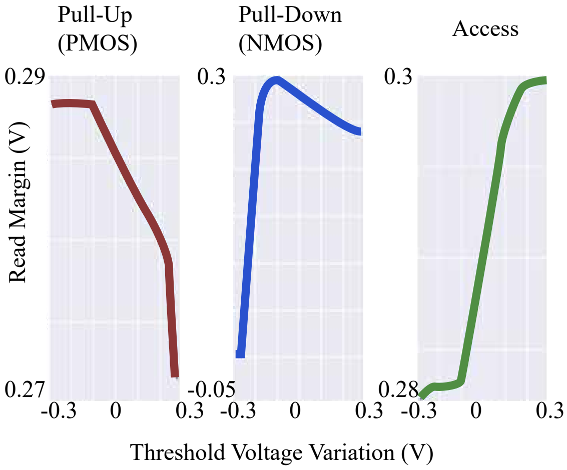


Figure 30: The sensitivity analysis of 6T dynamic read margin with variations in threshold voltages of PMOS (pull-up), NMOS (pull-down) and access transistor. All the three variations result in non-linear changes in the dynamic read margin.

The read margin is largely linear with respect to threshold voltage variation in the access transistor and non-linear for the others. A traditional approach would be to model this behavior in three dimensions by using a three dimensional smooth kernel,

$$K_{baseline} = \mathrm{RBF}([x_1, x_2, x_3], [x_1^*, x_2^*, x_3^*]) \tag{30}$$

where, $[x_1, x_2, x_3]$ are the threshold voltage variations for pull-up, pull-down and access transistors respectively. This kernel provides three-dimensional smooth functions as priors and can learn any three-dimensional continuous function given enough data. The disadvantage is that the learning becomes slow as dimensions increase, and it needs a larger number of input data samples than learning a one-dimensional function [Geenens et al., 2011]. Reducing the flexibility of the prior functions increases the learning rate. The flexibility of the prior functions for SRAM dynamic read margin can be reduced by using an additive kernel made of one-dimensional kernels, each modeling the read margin sensitivity with respect to individual threshold variation sources (one-dimensional RBF kernel for the pull-up and the pull-down transistors; one-dimensional linear kernel for the access transistors).

$$K_{additive} = \mathrm{RBF}(x_1, x_1^*) + \mathrm{RBF}(x_2, x_2^*) + \mathrm{Linear}(x_3, x_3^*) \tag{31}$$

Note that the $K_{additive}$ is sum of one dimensional kernels, while the $K_{baseline}$ is a three dimensional kernel. The interaction effect between these threshold voltage variation sources are then considered by adding them as product terms to the additive kernel.

$$\begin{aligned} K_{proposed} = RBF(x_1, x_1^*) + RBF(x_2, x_2^*) + Linear(x_3, x_3^*) \\ + RBF(x_1, x_1^*) * RBF(x_2, x_2^*) \\ + RBF(x_2, x_2^*) * Linear(x_3, x_3^*) \\ + Linear(x_3, x_3^*) * RBF(x_1, x_1^*) \end{aligned} \tag{32}$$

It should be noted that adding interaction terms increases the flexibility of the prior functions. These interaction terms increase exponentially with dimensions. Figure 31 provides the comparison in terms of mean in-sample error (with up to 400 training input samples) and mean out-sample error (with 10^4 test samples) for 20 iterations of the proposed model for read margin with the baseline model (three-dimensional RBF kernel) and other possible additive models. The out-sample error of a model decreases as the number of training samples increase because the model can then generalize better at the test points. The rate of decrease of out-sample error is defined to be the learning rate of the model. The In-sample error typically increases with more training sample because of the resulting increase in the squared error terms for the training samples. The preferred models are those with least out-sample error and faster learning rate [Abu-Mostafa et al., 2012].

Since the three-dimensional RBF kernel ($K_{baseline}$) provides the most flexible and smooth prior functions, its posterior functions have the minimum in-sample error in Figure 31. However the trade-off of this low in-sample error is the loss in the extrapolation capability as it is seen by its slow decreasing rate of out-sample error. Among the different additive models, the proposed model provides the least out-sample error. It also has the fastest learning rate among these models: after 200 training samples the out-sample error becomes less than 10^{-7}. In comparison, for the baseline model using three-dimensional RBF kernel, the out-sample error with 200 training samples is $7 * 10^{-5}$.

Also, it can be seen in Figure 31 that using the additive model with a one-dimensional RBF kernel for all three variation sources and their interaction also performs better than the baseline three-dimensional RBF kernel. However, this model has a higher out-sample error than our proposed model. This is because the prior functions generated by the one-dimensional RBF kernel (for the threshold voltage variations in access transistor) are more flexible than the linear prior functions to model the nearly linear read margin sensitivity as seen in Figure 30.

Thus, the results show empirically that for 6T bitcell and for the case of three threshold voltage variation sources, our memory margin model provides higher extrapolation than a model using three dimension universal kernel RBF. When we model SRAM margins for other bitcells such as 8T/10T, and assume the same threshold voltage variation sources (pull-up, pull-down and access) then memory hold (read) margin of these 8T/10T bitcells will be roughly twice the 6T bitcell read margin, because of their read bitline isolated design. However, the sensitivity trend (linear or non-linear) of their transistors remain the same as that shown in Figure 30 because these 8T/10T design have the same inverter structure and write path as the 6T bitcell. Because we are analysing the bitcells with the same number of variation sources, the dimension of the problem remains same. Hence the proposed model will again provide larger extrapolation accuracy than the three dimensional RBF kernel based model. In the other case, when we increase the variation sources to also include threshold voltage variation in buffer transistors, the dimension of the problem increases. The consequence of increasing the number of variation sources will be discussed in the next section, where we consider six threshold voltage variations (different threshold voltage variation in each transistor of 6T SRAM bitcell).

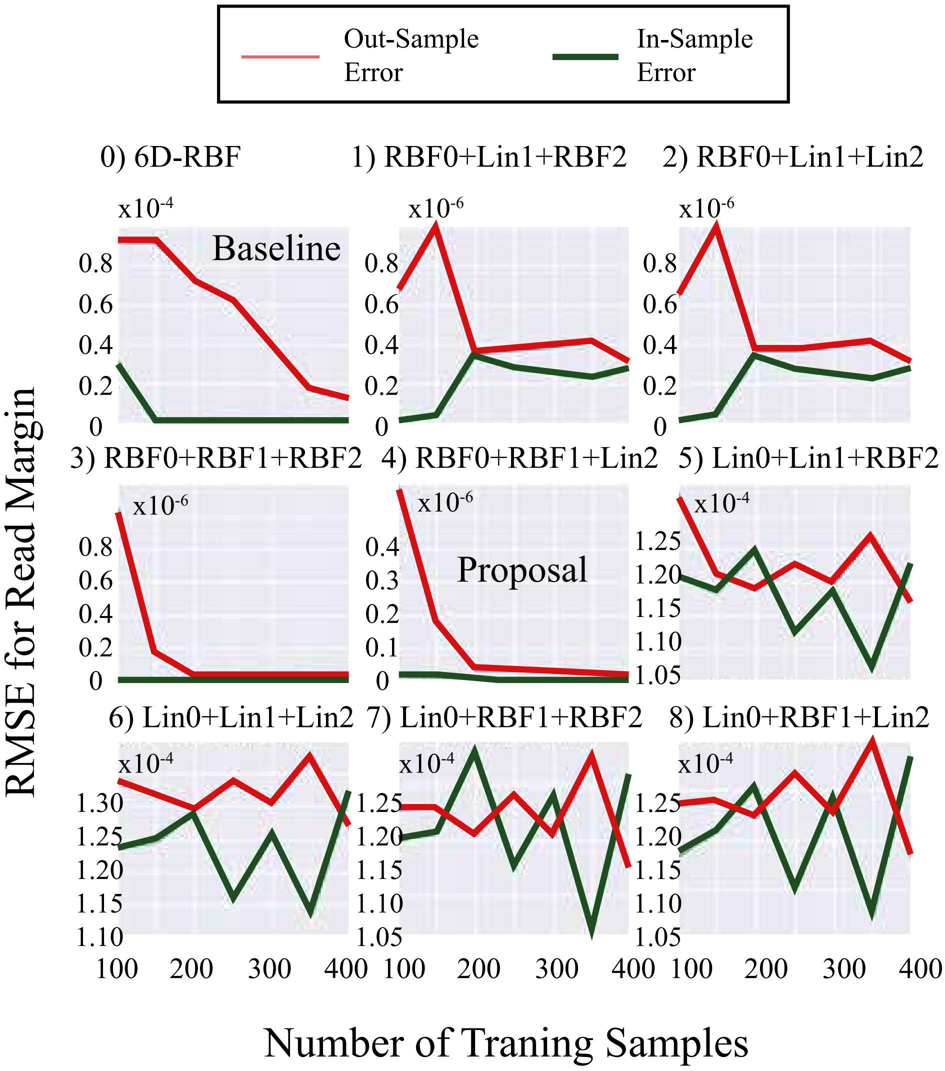


Figure 31: Mean Out-sample and mean In-sample error comparison for dynamic read margin modeling of 6T SRAM at sub-threshold voltage of 0.3V for 20 iterations. The blue line represents the Out-sample error and the green line represents the In-sample error. Since only a subset (10^4 samples) out of the entire input domain is taken, the Out-sample error shown in the figure is only an estimate of that over the entire input domain. The baseline for comparison is the error values for three-dimensional RBF kernel in sub-figure (0). The errors for proposed model [RBF+RBF+Linear+interaction terms] is shown in sub-figure (4). The proposed model gives the least out-sample error (prediction error for test samples) because increasing the structure information in kernels provide faster learning with smaller training samples than using a three-dimensional RBF kernel which has no structure information encoded.

4.4.2 Six dimensional Case

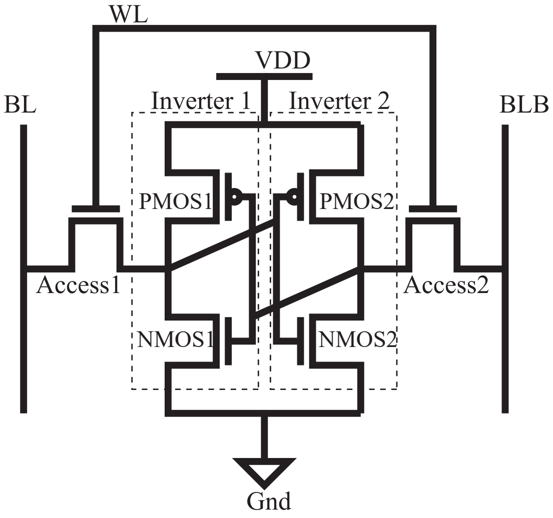


Figure 32: Schematic of 6T SRAM Cell.

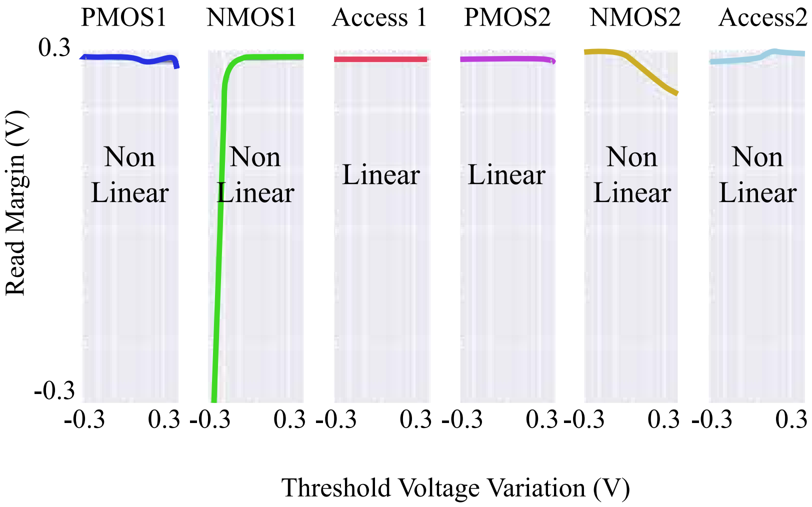


Figure 33: The sensitivity analysis of 6T dynamic read margin with variations in threshold voltages of PMOS (pull-up), NMOS (pull-down) and access transistors for each individual inverter. The inverter1 formed by [pmos1, nmos1 and access1] stores 1, while inverter2 formed by [pmos2, nmos2, access2] stores 0.

The sensitivity analysis of the dynamic read margin for a 6T SRAM cell (schematic shown in Figure 32) with respect to threshold voltage variations in the six transistors is shown in Figure 33. The inverter1 formed by [pmos1, nmos1, access1] stores 1 while inverter2 formed by [pmos2, nmos2, access2] stores 0. Thus, the bit-line noise from the access transistor affects more the inverter2 during read operation. When nmos1 has a high threshold voltage, the inverter1 stores a strong logic value 1 as the leakage current is reduced. When nmos1 has a low threshold voltage, the dynamic read margin decreases because of the leakage current. When inverter2 stores a logic 0, nmos2 with a high threshold voltage does not let the bit-line leakage current flow through the access2 transistor to pass through to ground, thereby raising the voltage at logic 0 and thereby decreasing the dynamic read margin. The sensitivity analysis shows that the read dynamic margin of the SRAM is non-linear with pmos1, nmos1, nmos2 and access2 threshold voltage variations. We model these with one-dimensional RBF kernels. While it is linear with access1 and pmos2 threshold voltage variations, we model these with one-dimensional Linear kernels. The baseline model for comparison is six-dimensional RBF kernel which can learn any continuous six dimensional function given enough data.

$$k_{baseline} = RBF([x_1, x_2, x_3, x_4, x_5, x_6], [x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*]) \tag{33}$$

Where, $x_1, x_2, x_3, x_4, x_5, x_6$ are Vth variations in transistors pmos1, nmos1, access1, pmos2, nmos2 and access2 respectively. This baseline model $K_{baseline}$, is a six dimensional model. Instead, we want to find a low-dimensional additive model similar to the two-dimensional proposed model in the previous section. We start with an additive model which is the sum of one dimensional kernels, each modeling the sensitivity of memory margin with respect to threshold voltage variation in one of the six transistors as seen in Figure 33.

$$k_{additive} = RBF(x_1, x_1^*) + RBF(x_2, x_2^*) + Linear(x_3, x_3^*) + Linear(x_4, x_4^*) + RBF(x_5, x_5^*) + RBF(x_6, x_6^*) \tag{34}$$

Since, this additive model is sum of one-dimensional model, it is also a one-dimensional model. There are 15 interaction effects that can be added to this additive model. However, not all of these interactions are present in the dynamic read margin for the 6T SRAM cell. For example, the interaction between transistor access2 in inverter2 and transistor access1 in inverter1 is not significant as both influence different storage nodes. Also the feedback loop present in the bit-cell nullifies any interaction between them caused by fluctuations in the storage node voltages. On the other hand, the transistor access1 and transistor nmos1 of inverter1 have a strong interaction effect on the read dynamic margin. In order to keep the model complexity to minimum, the following additive model is used which has interaction terms only between individual inverters:

$$\begin{aligned}
k_{proposed} = {} & (//\text{Additive terms}) \\
RBF(x_1, x_1^*) + RBF(x_2, x_2^*) + Linear(x_3, x_3^*) & \\
+Linear(x_4, x_4^*) + RBF(x_5, x_5^*) + RBF(x_6, x_6^*) & \\
(//\text{Interaction terms for Inverter1}) & \\
+RBF(x_1, x_1^*) * RBF(x_2, x_2^*) & \\
+RBF(x_2, x_2^*) * Linear(x_3, x_3^*) & \\
+Linear(x_3, x_3^*) * RBF(x_1, x_1^*) & \\
(//\text{Interaction terms for Inverter2}) & \\
+Linear(x_4, x_4^*) * RBF(x_5, x_5^*) & \\
+RBF(x_5, x_5^*) * RBF(x_6, x_6^*) & \\
+RBF(x_6, x_6^*) * Linear(x_4, x_4^*) &
\end{aligned} \tag{35}$$

We compare the mean in-sample (training input samples up to 1000) and mean out-sample error (for 10^6 test samples) of the proposed model for 20 iterations. For the proposed model (in Figure 34(1)), the increase in In-sample error with increasing training samples is less than the baseline (34(0)). The proposed model achieves lowest Out-sample error than all the other models. It has a faster learning rate than the rest, after 400 simulations, the Out-sample error converges to about $2.3 * 10^{-2}$ while for the six-dimensional RBF kernel the Out-sample error at 400 simulations is $3.6 * 10^{-2}$. Note that the error magnitude for six-variable case is larger than the three-variable case of previous section. This is true in general, the extrapolation accuracy of the model (both $K_{proposed}$ and $K_{baseline}$) decreases with the increase in the number of variables. Thus, in the

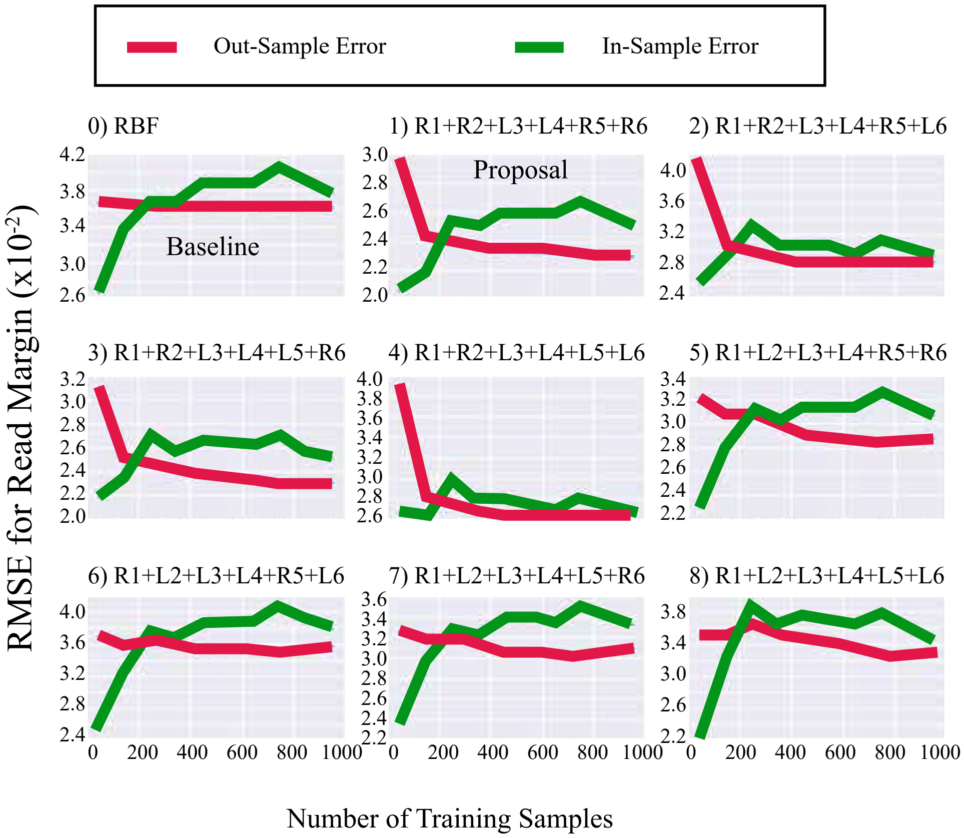


Figure 34: Mean Out-sample (prediction error for 10^6 test samples) and mean In-sample error (error in fitting of up to 1000 training samples) comparison for dynamic read margin modeling of 6T SRAM at sub-threshold voltage of 0.3V for 20 iterations. The blue line represents the Out-sample error and the green line represents the In-sample error. The proposed model shown in subfigure(1) gives the minimum Out-sample error.

case of 8T/10T bitcells, we have more variation sources than 6T bitcell. Hence, compared to the $K_{proposed,6T}$, the $K_{proposed,8T/10T}$ model (which now additionally includes the one-dimensional kernels modeling memory margin sensitivity with respect to buffer transistors and its interactions) will have lower extrapolation accuracy. In this case, former model is extrapolating data in six dimensional variable space, while latter model is extrapolating in eight/ten dimensional space . However, compared to the $K_{baseline,8T/10T}$ of 8T/10T, the $K_{proposed,8T/10T}$ will have higher extrapolation, where both models are extrapolating data in eight/ten dimensional space.

4.5 Dynamic Margin Failure Probability

The additive kernel described in the previous section was used to model the 6T SRAM cell's dynamic read margin. Since the model's learning rate (decrease in Out-sample error) for the proposed additive model does not increase significantly around 1000 training samples as seen in Figure 34, the initial sampling stage randomly samples 1000 training samples using Latin Hypercube Sampling (LHS) method [Pronzato and Müller, 2012] to ensure that each sampling space dimension is uniformly sampled. Figures 35a and 36a compare the margin values predicted by the additive model for 10^6 samples with the actual read and write margin values. Both the predicted and actual margin values are sorted in increasing order to make the comparison easier to visualize. The model fails to predict margin values below 0 (i.e. no Failure points). The reason is that there are not enough failure points in the training set to shift the predicted GP mean function below zero at the failure locations. Thus, in order to improve the failure predictions of the model the method is updated as follows:

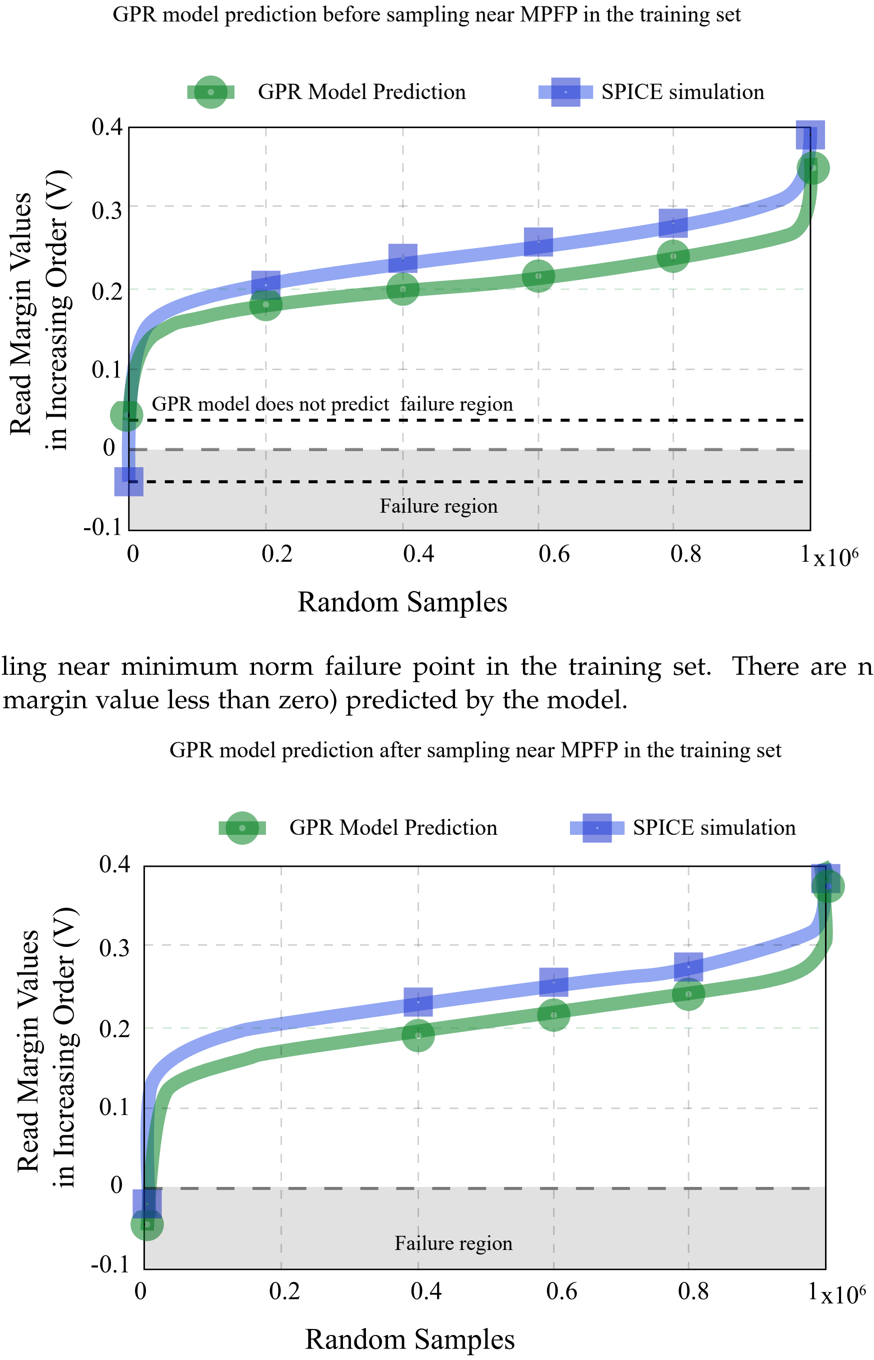
1. Training set of 1000 samples for the additive model is selected using Latin Hyper-cube Sampling (LHS).

2. The minimum norm failure point (MPFP) out of these training samples is selected.

3. If no failure points exists, then additional batch of 100 random samples (selected using LHS) are simulated. This step is repeated till a failure point is found. We sample within $\pm 3\sigma$ and distribution of failure samples among these 100 samples follow a binomial distribution (since we do not know the actual failure region we can only say that each sample is equiprobable to fail with probability equal to the failure probability of the memory margin). The probability of getting at-least a single failure sample from these 100 samples depends on the failure probability of the bitcell. For instance, in the case of bitcell failure probabilities 10^{-2}, 10^{-3} and 10^{-4}, the probability of getting at-least a single failure point among the 100 sampled points are 0.63, 0.09 and 0.009 respectively.

4. Add to the training set, 250 samples (1000/4) normally distributed around the minimum norm failure point (MPFP). The ratio 1/4th of the training set was empirically found to be the optimum ratio which did not over-estimate the predicted dynamic read margin failure region and thus prevent over-estimating read margin failure probability. This ratio may change for different supply voltages.

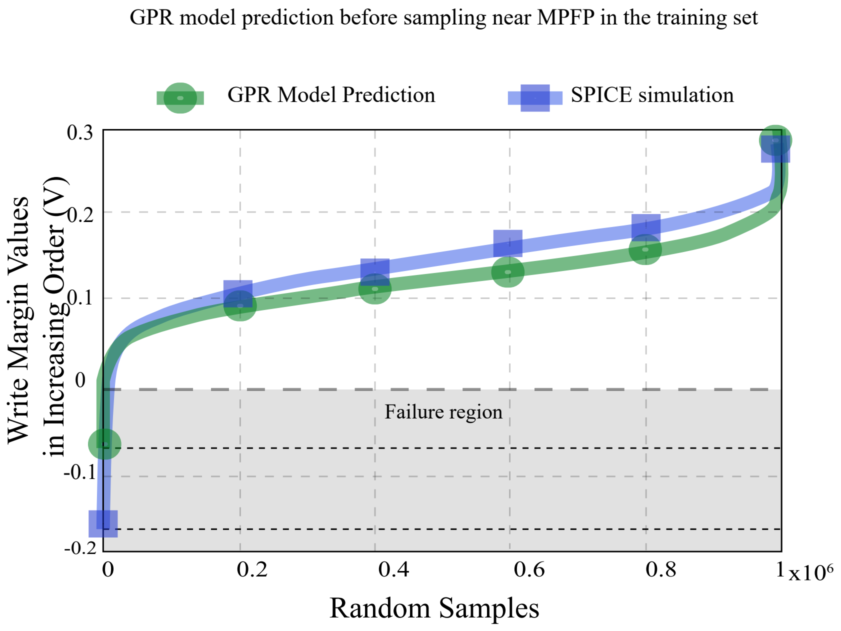
Figure 35b compares the predicted dynamic read margin values after adding 250 samples near the MPFP and its comparison with actual margin values for 10^6 test points. This predicted read margin by the additive model is then used as a surrogate model and then Monte-Carlo analysis is performed to estimate margin failure probability. This last step can also be performed using importance sampling on the surrogate model. Since the focus of the work is on reducing simulations to create a surrogate model and further sampling from the surrogate models is not computationally expensive, we have used a simple approach of using 10^6 Monte-Carlo simulations which can estimate failure probabilities up to 10^{-6}. The predicted values for read and write failure probabilities at 0.3V and 0.4V supply voltage, and their Monte-Carlo estimate are given in Table 3. The relative error of the predicted dynamic read margin at 0.4V supply voltage is 30% compared to its Monte Carlo estimate. The maximum relative error is 210% for dynamic write margin at 0.3V. The relative error is larger for higher failure probability values because same number of 250 points are sampled near MPFP in step 4 in both the cases. The fraction of these samples which are failure points is higher in the case $5.4 * 10^{-3}$ failure probability and as such the method overestimates the failure probability to $1.7 * 10^{-2}$. This approach can be improved by using generalized Pareto distribution (GPD) to accurately fit the tail of the dynamic margin



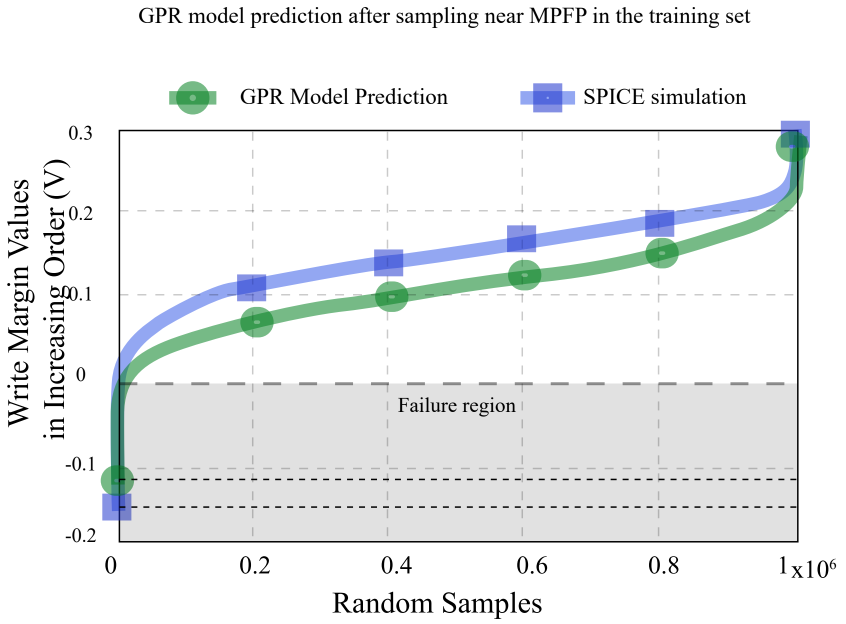
(a) Before sampling near minimum norm failure point in the training set. There are no failure points (points with margin value less than zero) predicted by the model.

(b) After sampling additional points near minimum norm failure point in the training set. The model is able to predict failure points.

Figure 35: Read dynamic margin values at 0.3V predicted by the additive kernel model with 1000 input training samples and compared with the actual margin values for 10^6 points on the input space. Both the predicted and actual margin values are sorted in increasing order.



(a) Before sampling near minimum norm failure point, fewer failure points are predicted by the model.



(b) After sampling additional points near minimum norm failure point the model is able to predict failure points.

Figure 36: Write dynamic margin values at 0.3V predicted by the additive kernel model with 1000 input training samples and compared with the actual margin values for 10^6 points on the input space. Both the predicted and actual margin values are sorted in increasing order.

Table 3: Predicted dynamic read margin and write margin failure probabilities

Method	Dynamic Margin	#Spice Simulations	Estimated Failure Probability
Monte Carlo	Read Margin @ 0.3V	10^6	$1.1 * 10^{-5}$
Proposal	Read Margin @ 0.3V	1250	$3 * 10^{-5}$
Monte Carlo	Write Margin @ 0.3V	10^6	$5.4 * 10^{-3}$
Proposal	Write Margin @ 0.3V	1250	$1.7 * 10^{-2}$
Monte Carlo	Read Margin @ 0.4V	10^6	$3 * 10^{-6}$
Proposal	Read Margin @ 0.4V	1250	$4 * 10^{-6}$
Monte Carlo	Write Margin @ 0.4V	10^6	$6.3 * 10^{-4}$
Proposal	Write Margin @ 0.4V	1250	$1.2 * 10^{-3}$

distribution, as proposed in [Wu et al., 2014]. The failure regions can be classified using proposed additive kernels for Gaussian process instead of using the Gaussian Radial Basis kernel (GRBF) base vector machine (SVM) [Wu et al., 2014]. The comparison of accuracy given in [Zhao et al., 2015] for predicting the Monte Carlo estimate of $2.3 * 10^{-4}$ with REscope[Wu et al., 2014] and recursive statistical blockade [Singhee and Rutenbar, 2008], shows relative error between 20% and 64%. Thus proposed method provides similar accuracy numbers (minimum relative error of 30%) with speed-up in computation between 4x and 23x compared to these previous methods.

4.6 Conclusion

In this chapter, we showed that for modeling SRAM dynamic margins, the extrapolation error (out-sample error) can be decreased by using additive kernels encoding the structure present in the sensitivity analysis of the dynamic margin functions, instead of using universal kernels such as six-dimensional RBF. We presented the case of modeling dynamic read margin as an example for the efficacy of additive models made by using one-dimensional kernels and their interactions as sum of product kernels. The response surface generated by Gaussian process using these proposed models is then used to estimate failure probabilities with 1250 simulations. These predicted failure probability values are then compared with Monte-Carlo analysis with 10^6 samples and show relative error of 1.72 for dynamic read margin at 0.3V supply voltage.

Other kids' baseball heroes hit home runs. Mine gets sent down to the minors!

Charlie Brown, Peanuts

5

10TSD:Near threshold Bitcell for Faster Read Access

5.1 Introduction

The prevalence of portable computing, made possible by the presence of battery operated mobile computing devices, has raised interest in energy efficient computing. These battery operated low power systems for the 'Internet of things' are being used in domains ranging from simple house monitoring to complex industrial monitoring systems [Bol et al., 2013]. The life of these systems is limited by the lifetime of the battery. It is not possible to just use larger batteries as that decreases the portability of these devices. The only way forward is to make the system more energy efficient. This energy efficiency can be achieved at supply voltages near the sub-threshold level [Calhoun and Chandrakasan, 2004].

A big challenge for sub/near-threshold operation to become reality is the loss of SRAM operation at such low voltages. Hence, subthreshold SRAM design has emerged in the last few years. It aims at providing on-chip memory system capable of operating below 0.4V supply voltages. Nevertheless, this comes at the cost of a reduction in the operation speed and a higher failure probability due to variations [Boley et al., 2012],[Raychowdhury et al., 2005]. The presence of short channel effects (which increases as we move to lower technology nodes) is also shooting up the probability of failure of the traditional bitcells. SRAM bitcells are, thus, designed with strict sizing calculations and they include topology modifications to decrease their failure probability. This situation worsens near the sub-threshold operation. The Read Static Noise Margin of a bitcell is dependent on the ratio of the drive currents of the pull down and the pass transistors. In subthreshold operation, subthreshold current acts as the drive current of a transistor and it is exponentially dependent on its threshold voltage. Hence, the read SNM of a bitcell in subthreshold operation becomes exponentially dependent on the threshold voltage variations in the pull down and pass transistors. To increase the robustness of a bitcell at subthreshold voltages, one can increase the size of its transistors. However this approach will increase the energy per operation. Consequently, naive upsizing of transistors goes against the motivation to use subthreshold voltages which is to achieve maximum energy efficiency. As a consequence, structural modifications have become a necessity. The subthreshold bitcells proposed in recent years have focused only on increasing the stability while there has been less work on reducing the delay of these bitcells. Thereby, the use-case of subthreshold operation has mostly been limited to low performance driven domains like wireless sensor networks [Calhoun et al., 2005]. There is a need for SRAM bitcells that can extend the energy benefits of subthreshold operation to mid-performance domains like mobile computing.

In this chapter we propose a buffer based bitcell, 10TSD, that achieves similar stability as 10T bitcell and has only half the read delay as of 10T. This is made possible by using a new buffer structure where the discharge path consists of only a single transistor.
The chapter is organized as follows: Section 5.2 provides a brief overview of the recently proposed bitcells and the modifications they make to achieve reliable operation. Section 5.3 introduces the proposed bitcell. Section 5.4 presents the results of the HSPICE simulations of the bitcell. Finally, section 5.5 draws the conclusions.

5.2 Related Work

The sub-threshold SRAM bit cells proposed in recent years have adopted one or more of the following upgrades over the traditional 6T bit cell.

- To provide a discharge path for bitlines isolated from internal storage nodes. E.g. the traditional 8T bitcell [Verma and Chandrakasan, 2008], the subthreshold 8T bitcell proposed in [Kim et al., 2007b], and the 10T bitcells proposed by Kim et al [Kim et al., 2011] Figure 37 and Calhoun et al [Calhoun and Chandrakasan, 2006a] Figure 38. The Read SNM of these bitcells is similar to the Hold SNM of the 6T bitcell. The Hold SNM of the 6T bitcell at 0.3V is the same as its Read SNM at 0.6V supply voltage; and thus, these bitcells have larger noise margins than a 6T bitcell at subthreshold supply voltages. Larger area and slower read access are the main issues with this approach.

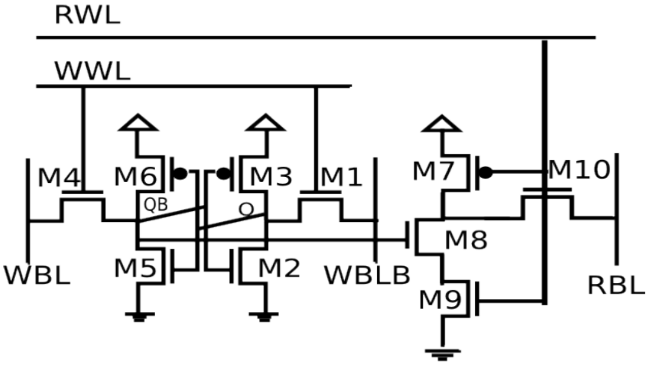


Figure 37: 10T-Kim single ended buffer based SRAM bitcell [Kim et al., 2011].

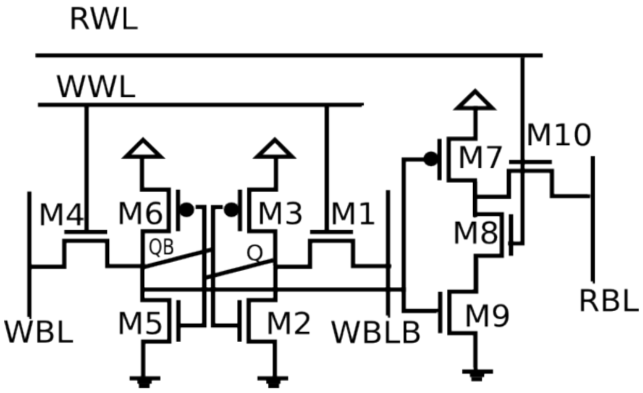


Figure 38: 10T-Calhoun buffer based SRAM bitcell [Calhoun and Chandrakasan, 2006a]

- To provide a pseudo storage node by adding an extra pull-up or pull-down transistor to the inverter structure. The bitline discharge path goes through this pseudo-storage node instead of the actual storage node. Thus, the bitline noise does not interfere with the value stored in the actual storage nodes. E.g. the 9T subthreshold bitcell proposed in [Chang et al., 2011] The bit cell has the read bitline connected to a pseudo storage node which is disconnected from the internal storage node during read operation. The motivation for these bitcells is also to reduce the current influx into the internal storage node. The main drawback of these bitcells is the modification of the inverter threshold due to the addition of the extra transistors in the inverter structure.

- To provide a feedback cut-off mechanism between the two inverters. Eg. the 9T bitcell proposed in [Chang et al., 2012] This cell tries to increase the write-ability of the cell by breaking the feedback between the inverter loop. Other similar bitcells break the feedback during read operation so that one of storage nodes retains the correct value even if the other storage node inverter flips.

- To provide a hysteresis effect to the inverter state transitions to force a higher threshold voltage in the 0 to 1 logic state transition. Thus, the threshold voltage of the inverter storing '0' is higher than the other and it requires a larger voltage drop on the node storing '1' to flip a node storing '0'. eg. Schmitt Trigger based bitcell [Kulkarni and Roy, 2012].

Our proposed bitcell is an isolated read bitline design. In that way, it is similar to the buffer based 8T and 10T bit cells. However, it has a single transistor in the bitline discharge path and, in that way, it is different from these buffer based designs. The two inverter feedback structure however is kept as the original 6T cell. Thus, our proposal does not have the overhead associated to other proposals such as the pseudo-storage node, the feedback cutoff and hysteresis bitcells that add extra transistors to the inverter feedback structure.

5.3 Proposed 10T Bitcell

Bitcell Schematic

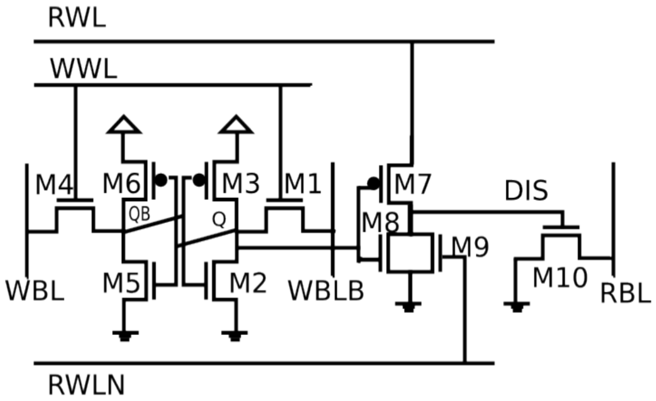


Figure 39: The schematic of Proposed 10T Bitcell with single transistor Read-Bitline discharge path.

The schematic of the proposed bitcell is shown in Figure 39. The inverter-feedback and the write structure of the bitcell (transistors M1 to M6) are the same as in the traditional single ended buffer-based bitcells and the 6T bitcell. The read buffer consists of one PMOS (M7) and three NMOS (M8, M9 and M10) transistors. The output of the structure formed by M7, M8 and M9 (node DIS) is used to drive the read-bitline (RBL) discharging transistor M10. During the IDLE stage, when the read-wordline(RWL) is driven low, the read-wordline-complement (RWLN) is driven high. The node DIS discharges to the ground. As a consequence, transistor M10 is turned off when the bitcell is not being read.

Having only a single transistor in the RBL discharging path offers two advantages against the traditional discharge path of two stacked NMOS transistors:

- The capacitive load on the read bitline is smaller; the read bitline swing is achieved by spending smaller dynamic energy.
- The resistance in the discharge path is smaller; the on-current (I_{on}) while discharging the read-bitline is higher which results is smaller delay.

However, this advantage comes at the cost of higher leakage currents (I_{off}) from the read-bitline, as we will see and quantify in Section 5.4.

Bitcell Read operation

When the read cycle begins, the RWL is driven high (RWLN is thus driven low). M7 and M8 together form an inverter with storage node Q as input and node DIS as output. (RWLN is driven high only after the read operation finishes and until the node DIS discharges, otherwise RWLN stays driven low)

If Q stores logic value '1', the inverter (M7,M8) output DIS is low. The transistor M10 does not discharge the RBL. Thus, when reading the logic value '0', the RBL is not discharged. Only off-current (I_{off}) flows from the bitline through M10.

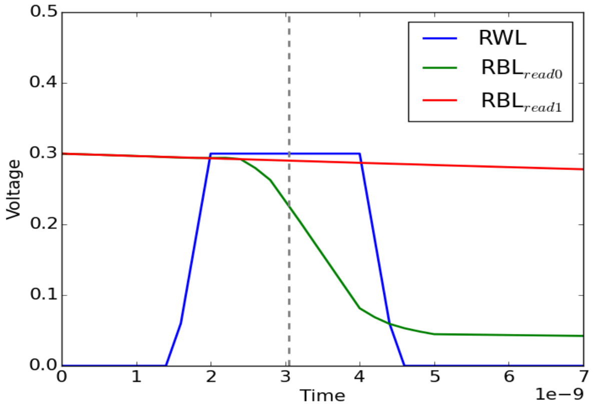


Figure 40: Read '0' and Read '1' operation of bitcell @0.3V supply voltage.

If Q stores the logic value '0', the node DIS is high and transistor M10 discharges the RBL with on-current I_{on}. After the read operation is finished, the RWLN turns on M9 to discharge the node DIS back to '0'.

Figure 40 shows the read '0' and read '1' operation of the bitcell at sub-threshold supply voltage 0.3V. In the Figure 41, the transition in the voltage at node DIS during read '0' shows that the V(DIS) is driven low after the read operation finishes.

During the idle stage, M10 transistor is connected to virtual-ground to prevent the leakage from Read-Bitline.

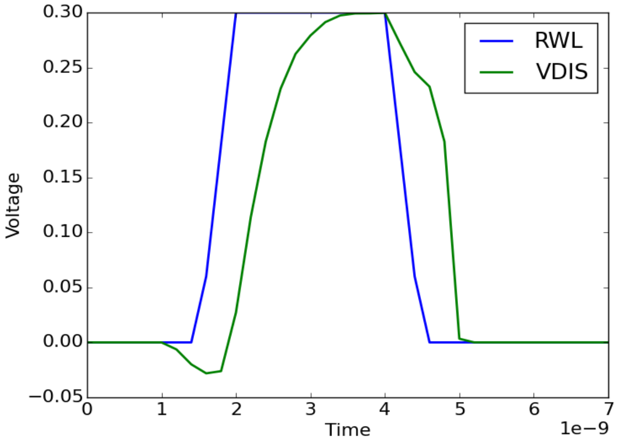


Figure 41: The charging (during the Read ′0′) and discharging (after read) of the node DIS.

Bitcell Write Operation

The write operation is similar to the traditional single ended buffer based designs. The Write-Wordline is driven high with new data at the Write-Bitlines (WBL and WBLB). The Write access transistors (M1 and M4) write the new logic values available in the Write-Bitlines into the storage nodes (Q and QB). Figure 42 shows the transient wave-forms of both the Write ′0′ and Write ′1′ operation.

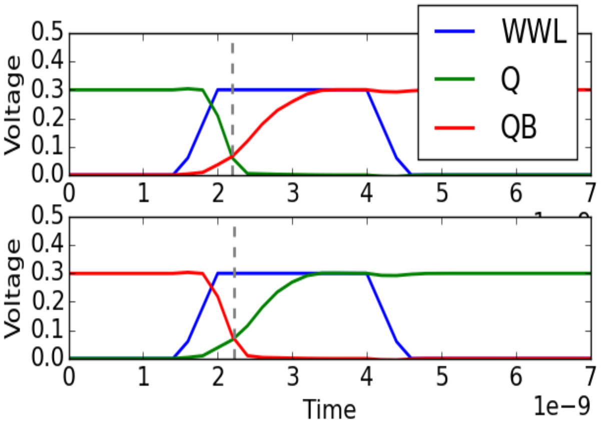


Figure 42: Write ′0′ and Write ′1′ operation @0.3V. Write Delay ≈ 0.7ns.

Area Comparison

The layouts of the bitcells using the design rules for 65nm are shown in Figures 43, 44 and 45. The initial study on the area of the bitcells shows that the proposed bitcell, 10TSD, has 13.6% larger area than 10T-Kim and 7% larger area than 10T-Calhoun due to the increase in layout width by PMOS transistor M7.

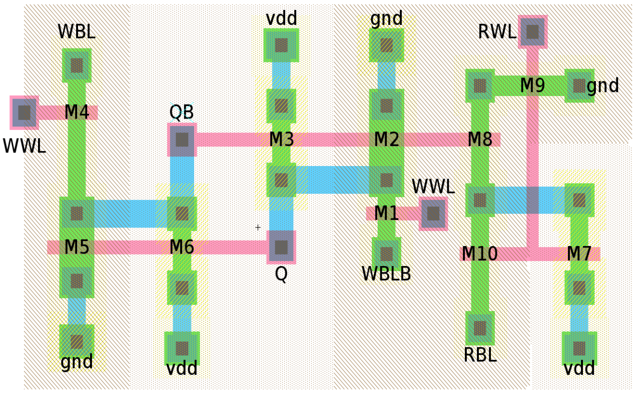


Figure 43: 10T-Kim SRAM bitcell layout

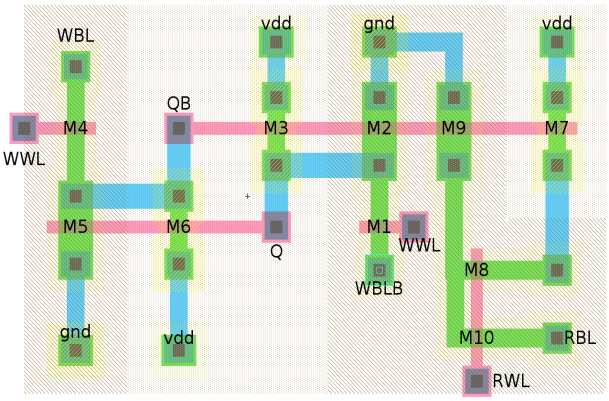


Figure 44: 10T-Calhoun SRAM bitcell layout

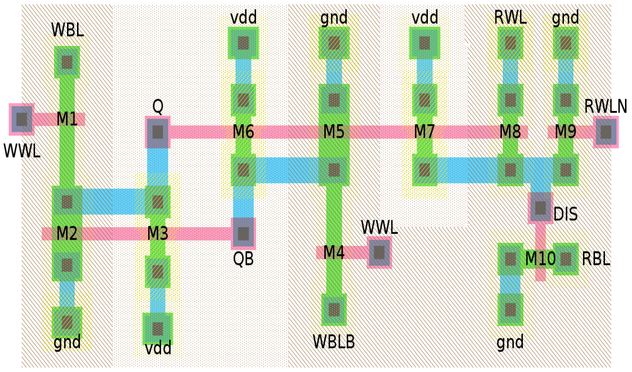


Figure 45: Proposed 10TSD SRAM bitcell layout

Table 4: Transistor Widths (nm)

Transistor	Width (nm)
M1	100
M2	200
M3	100
M4	100
M5	200
M6	100
M7	100
M8	100
M9	100
M19	100

5.4 Performance Comparison Results

Methodology

The spice netlist of bitcells is simulated in HSPICE using the 65nm, 45nm and 32nm PTM model with widths specified in Table 4. Variability is simulated as a random threshold voltage fluctuation. The threshold voltages of the NMOS and PMOS transistors are modeled as Gaussian distributions with mean around their technology values and sigma proportional to the sizes of each transistor. 1000 Monte-Carlo simulations are run for the variability analysis of energy and delay. The dynamic stability is estimated using Importance Sampling with 5K random samples to estimate the failure probabilities. The analysis is done at sub/near-threshold voltage range of [0.3V,0.4V,0.5V] and column sizes of [32, 64 and 128 bits/column]. The performance metrics (energy and delay) of the proposed bitcell are compared with the 10T-Kim [Kim et al., 2011] (referred as 10T-1 in figures) and 10T-Calhoun [Calhoun and Chandrakasan, 2006a] (10T-2 in figures).

For transient analysis, the read '0' delay is defined as the time starting from the rise in Read-Wordline (0.5*Vdd) till the time it takes for the read-bitLine (RBL) to drop to 0.5*Vdd. The read energy is computed as the energy spent while the read-wordline (RWL) is high. The dynamic energy is the energy spent by 'ON' transistors and leakage energy is the energy spent by 'OFF' transistors. For instance, in Read-0 case for the proposed bitcell, dynamic energy is the summation of energy spent by transistors M2, M6, M7 and M10, while leakage energy is the energy spent by rest of the transistors. The RWL pulse width used is 8ns. Ion/Ioff ratio is defined as the ratio of the read-0 current of the accessed bitcell to the current from the non-accessed bitcells in a column.

Read Delay Comparison

At ultra-low voltages, the RBL in the 10T-Kim and 10T-Calhoun bitcells during the read '0' operation is unable to discharge to '0.5*Vdd' for smaller Read-Wordline (RWL) pulse widths of 2ns (Figure 46, 47, 48). Increasing the number of bitcells per column results in a read access failure for these 10T bitcells. At 0.4V supply voltage, the proposed bitcell provides a speedup of 2.3x with 64 bitcells/Column and 3x with 128 bitcells per column. In presence of local variations, the $\mu + 3\sigma$ delay is 12%,49% and 34% smaller at 0.3V,0.4V and 0.5V for a column size of 32-bits as shown in Figure 49.

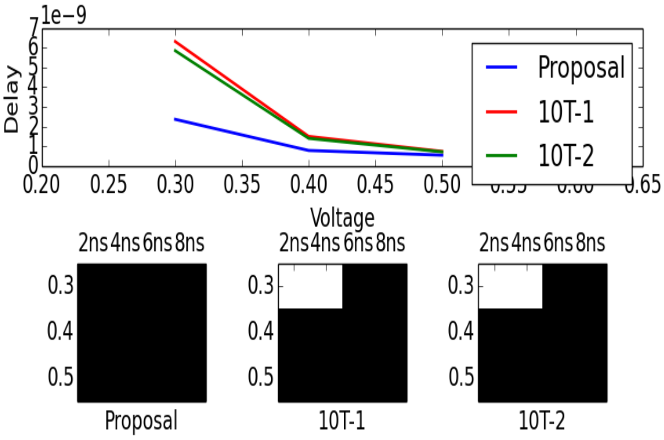


Figure 46: Read '0' delay for 32 bitcells/column. Speedup in access time is 2.4x(@0.3V), 1.8x(@0.4V), 1.3x(@0.5V) and 1.1x(@0.6V). Bottom figures show the read access failures (white) or not(black) for different read-wordline pulse widths. The proposed bitcell does not have any failures while the 10Tkim(10T-1) and 10TCalhoun(10T-2) fail for $<$ 0.4V supply voltages at 2ns RWL pulse width.

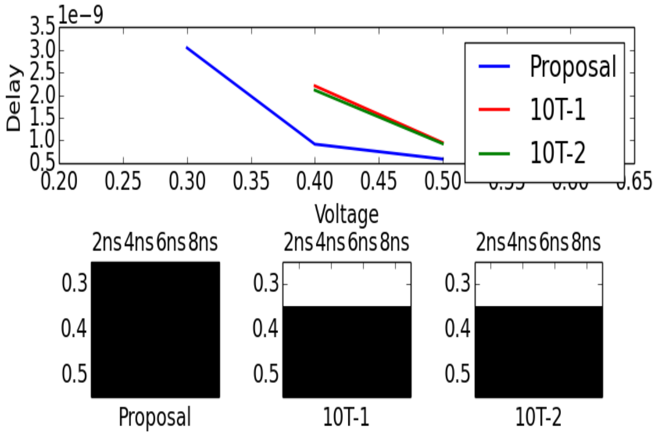


Figure 47: Read '0' delay for 64 bitcells/column. Speedup in access time is 2.3x(@0,4V),1.5x(@0.5V) and 1.2x(@0.6V). For a short read-wordline pulse width of 2ns both 10Tkim(10T-1) and 10TCalhoun(10T-2) fail in the read operation.

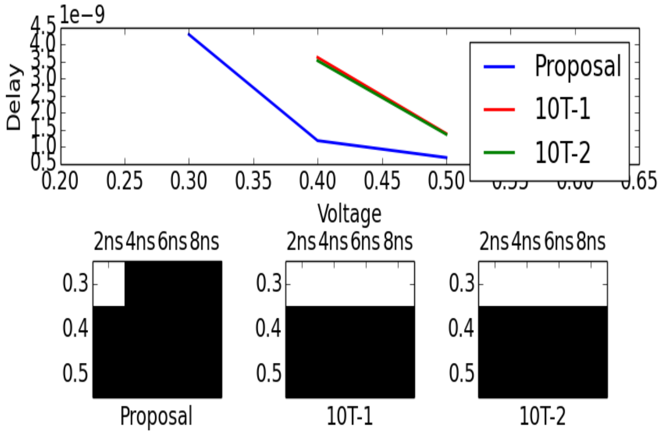


Figure 48: Read '0' delay 128 bitcells/column. Speedup in access time is 3x (@0.4V), 2x(@0.5V) and 1.57x(@0.6V). At 0.3V both 10T bitcells 10T-Kim (10T-1)and 10T-Calhoun(10T-2) fail in the read operation, while for the proposed bitcell only fails at 0.3V with a 2ns RWL pulse.

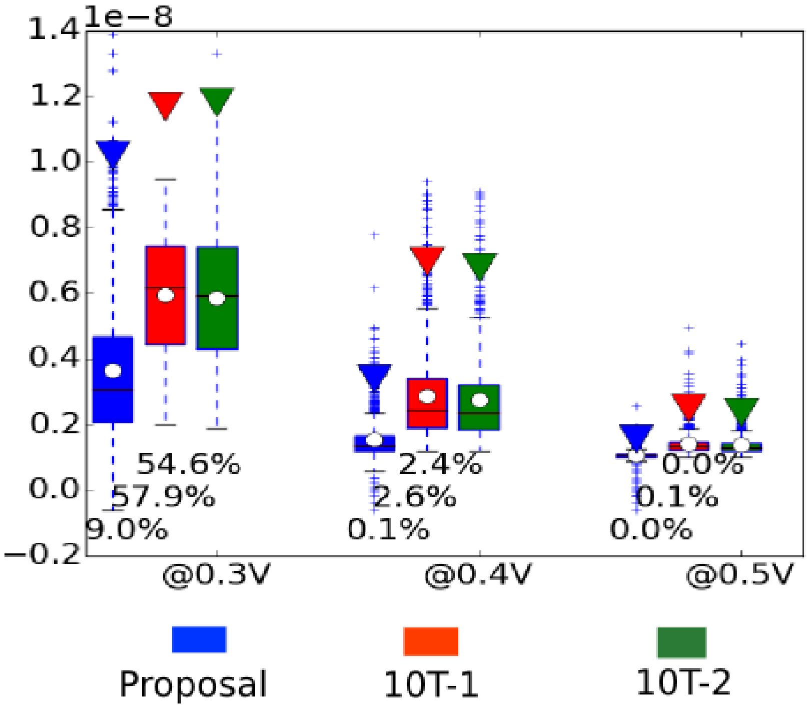


Figure 49: Boxplot for Read-0 delay under local variations for 32bits/column. Mean is denoted by the circle and the wedge denote the $\mu + 3 * \sigma$ value. The percentages show the %read-access failures (Bottom % is for our proposal, the middle is for the 10T-1 and the topmost is for the 10T-2)

The single transistor discharge path suffers from a low Ion/Ioff ratio (ratio of the read-0 current to the current from non-accessed bitcell in a column), as shown in Figure 50 the median Ion/Ioff decreases as bits/column are increased . The median Ion/Ioff ratio without virtual ground at 0.3V is 38 (32bits/column), 19 (64bits/column) and 9 (128bits/column), and at 0.4V is 205 (32bits/column), 103 (64bits/column) and 51 (128bits/column). For comparison, the Ion/Ioff ratio for buffer based bitcells at 0.4V is $\geqslant 10^3$, as seen in Figure 51. A better Ion/Ioff ratio can be attained by preventing the flow of leakage current from the bitline by the non-accessed memory cells in the column, and for that M10 is connected to a virtual ground when its Read-Wordline is driven low. Figure 50 shows the improvement in the Ion/Ioff which reaches a median value of 10^4 for the 32bits/column configuration. The median Ion/Ioff ratio in presence of local variations for the proposed bitcell is $\approx$ 6 times that of the other bitcells at 0.3V and 0.4V and $\approx$ 4 times at 0.5V, as seen in Figure 51 and Figure 52.

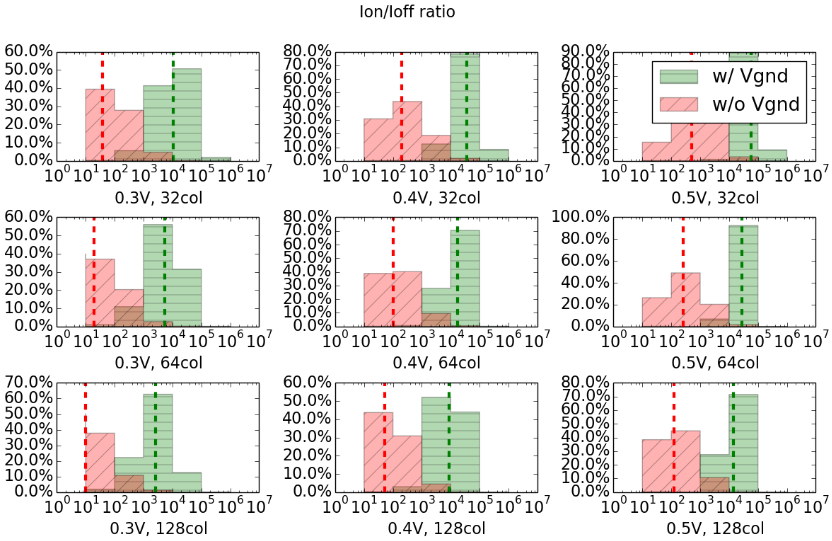


Figure 50: Ion/Ioff ratio under local process variations for the proposed cell with 32bits/column, 64bits/-column and 128 bits/column at 0.3V, 0.4V and 0.5V

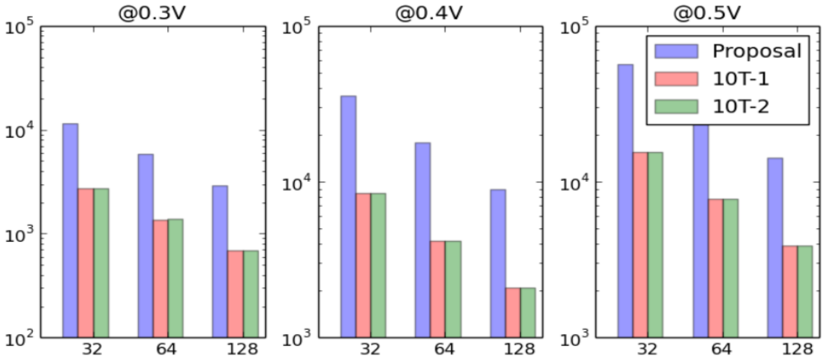


Figure 51: Ion/Ioff ratio comparison for 32, 64 and 128 column sizes at a) 0.3V b) 0.4V and c) 0.5V

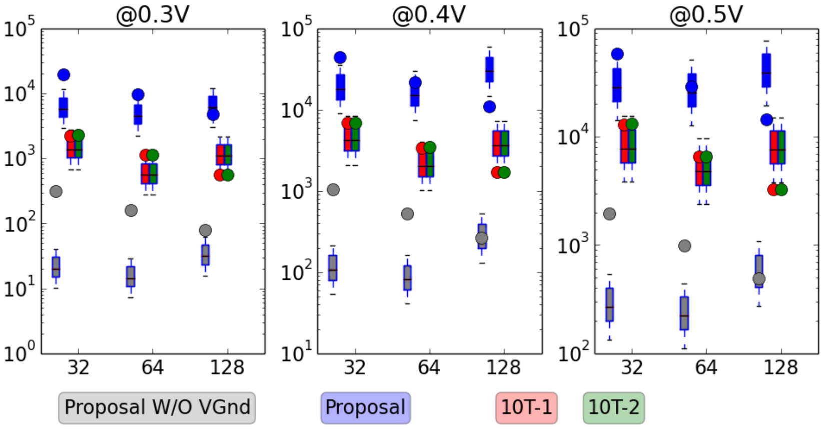


Figure 52: Ion/Ioff ratio under variations, circle denotes the $\mu + 3\sigma$ value while horizontal line inside boxes denotes the median value

Read Energy Comparison

During the read '1' operation, the proposed bitcell has only one transistor on (i.e M8 in Figure 39) compared to two and three transistors in 10T-Kim and 10T-Calhoun for read-0, read-1 and non-accessed bitcell in a column). In the proposed bitcell, RWLN is driven low during read and hence only RWL contributes to word-line energy consumption for read access. Thus the proposed bitcell spends far less dynamic energy during the read '1' operation than the other 10T bitcells. During the read '0' operation also two transistors are turned on compared to three transistors in the buffer of 10T-Calhoun. Figure 53 shows the energy comparison of the three 10T bitcells. The effect of local process variations on the distribution of energy is shown in Figure 54 and Figure 55 where $\mu + 3\sigma$ are compared.

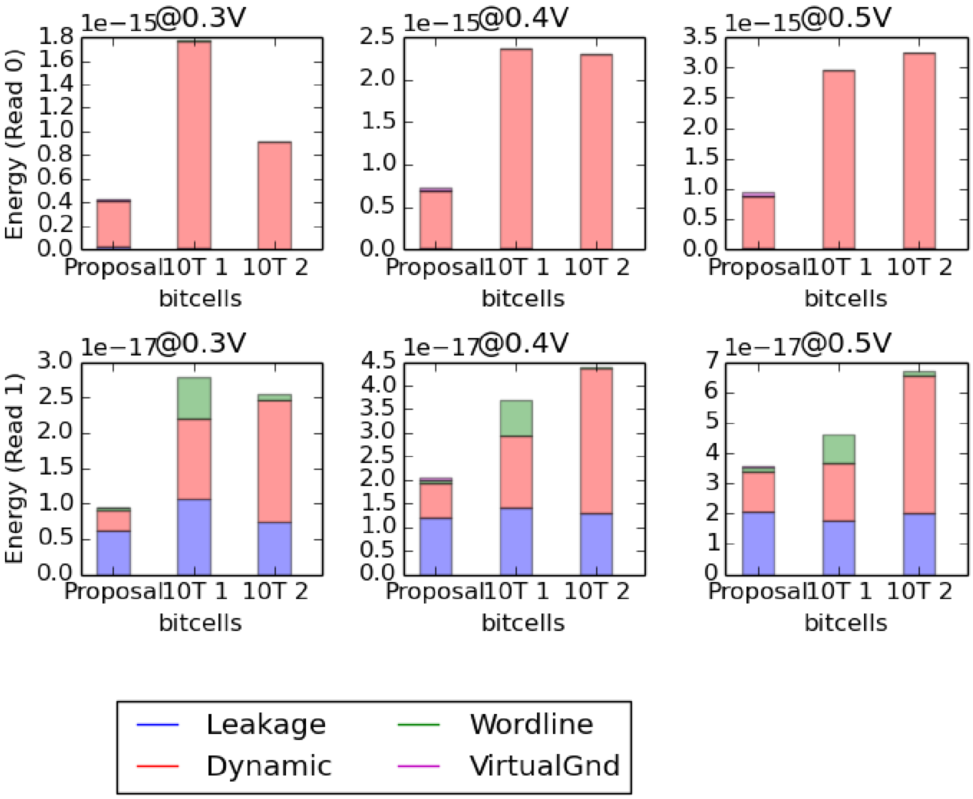


Figure 53: Energy comparison for Read '0' and Read '1' operation of bitcell @0.3V,0.4V and 0.5V supply voltage.

The percentage increase/decrease in leakage, dynamic and total energy of proposed bitcell vs the 10T bitcells are compared in the Tables 5 and 6. The fourth column in these tables ("Proposal Leak/Total") is the % contribution of proposal's leakage energy to its total read-energy at 0.3V,0.4V and 0.5V. For Read-0 it is very small percentage. So, even though there is larger increase in leakage, it is overshadowed by the decrease in dynamic energy because of leakage's smaller percentage contribution to the total read-energy. Hence there is an overall decrease in total read-energy. The RWLN is driven high only after the read-access until it discharge the node DIS which takes less than 1ns (as seen in Figure 41). The increase in the energy because of RWLN for the proposed bitcell is only 1% when RWLN is driven high for 1ns at 0.4V.

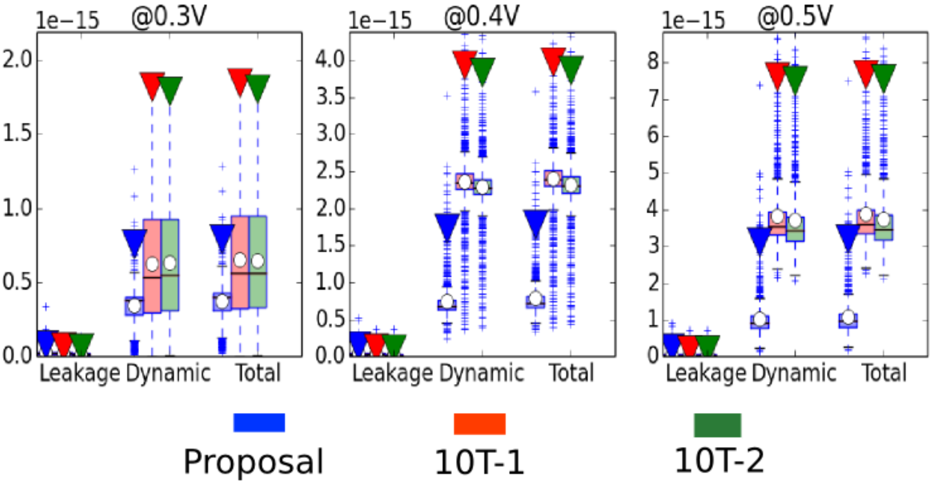


Figure 54: Boxplots for Read-0 Leakage, Dynamic and Total (Leakage+Dynamic) Energy under local variations. Mean is denoted by the circle and the wedge denote the $\mu + 3 * \sigma$ value. Line within Boxes denote the median value. Variation in total Read-0 energy is minimum for the proposal.

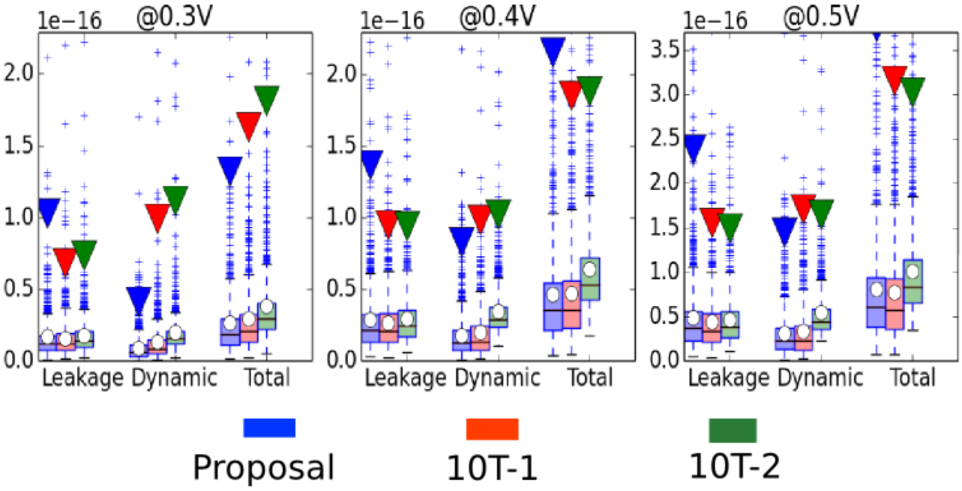


Figure 55: Boxplots for Read-1 Energy under local variations. Variation in leakage is higher for the proposal.

Table 5: READ-0 ENERGY COMPARISON

	ΔLeak	ΔDyn	ΔProposal Leak/Total	ΔTotal	$\mu + 3\sigma$	
0.5V	42%↑	70%↓	2.56%	69.9%↓	58%↓	vs 10T-1
	30%↑	73%↓		72.2%↓	57%↓	vs 10T-2
0.4V	101%↑	70%↓	3.44%	70% ↓	54%↓	vs 10T-1
	137%↑	70%↓		69.5%↓	53%↓	vs 10T-2
0.3V	61%↑	71%↓	4.78%	69.8%↓	56%↓	vs 10T-1
	235%↑	57%↓		54.9%↓	55%↓	vs 10T-2

Table 6: READ-1 ENERGY COMPARISON

	ΔLeak	ΔDyn	ΔProposal Leak/Total	ΔTotal	$\mu + 3\sigma$	
0.5V	17%↑	30%↓	59%	24%↓	19%↑	vs 10T-1
	2%↑	70%↓		47%↓	24%↑	vs 10T-2
0.4V	17%↑	32%↓	60%	13% ↓	16%↑	vs 10T-1
	5%↓	76%↓		54%↓	14%↑	vs 10T-2
0.3V	8%↑	47%↓	66%	28%↓	18%↓	vs 10T-1
	15%↓	83%		63%↓	26%↓	vs 10T-2

Thus, the proposed bitcell provides a minimum of 54% reduction in total energy for Read-0 and 24% reduction in total energy for Read-1 by trading-off leakage energy for dynamic energy. When evaluating leakage, not only it is important to analyze the leakage current magnitude (as reported in this work) but to bear in mind that the leakage energy also depends on the execution time of the application and operating voltage. Since the proposed bitcell can read at higher frequencies even at very low voltages (i.e. 0.3V), it allows the operation at very low voltages with a higher processor frequency and, thereby, reducing the execution time and most possibly the overall leakage and energy consumed. In addition, with other technologies (i.e. FinFETs) likely to replace planar-CMOS, the reduction of leakage energy they provide will result in possibly even better energy reduction and even higher Ion/Ioff ratios.

Read and Write Stability Comparison

The read/write dynamic stability of the bitcells are compared in Tables 7 and 8 . The failure probabilities are estimated with 5000 Monte-Carlo simulations with variation in the threshold voltage of each transistor. Each of these bitcells is simulated with the same 5K random samples for threshold voltage variation. The 10T-1, 10T-2 and the proposed bitcell 10TSD are read-bitline isolated designs, hence their read stability is the same as their hold stability. Tables 7 and 8 show that these bitcells have similar dynamic-stability for both read and write operation. This is expected because the hold/write stability of these bitcells is only dependent on the stability of the feedback inverter structure (M2-M3 and M5-M6). This feedback inverter structure is same for each of these bitcells. The same write path results in similar stability for write operation. The RWL has no effect on the storage node (Q and QB) voltages because of the read-bitline isolated design, hence their Read(Hold) stability is also similar.

Table 7: Dynamic read failure probability

	@0.3V	@0.4V	@0.5V
Proposal	0.015	0.0015	0.001
10T-1	0.0165	0.0015	0.001
10T-2	0.0155	0.0015	0.0008

Table 8: Dynamic write failure probability

	@0.3V	@0.4V	@0.5V
Proposal	0.79	0.589	0.355
10T-1	0.773	0.589	0.355
10T-2	0.794	0.589	0.355

Write Delay and Energy Comparison

The write operation of these 10T bitcells is same as the traditional 8-transistor SRAM bitcell. The write-wordline is driven high with data on the write bitlines. The write operation is symmetric, that is write ′1′ operation at storage node Q is same as write ′0′ operation at storage node QB. Since the write path is same for each of these 10T-bitcells, they are expected to have similar write delay and write energy as seen in Tables 9 and 10.

Table 9: Write Delay

		Proposal	10T-1	10T-2
@0.3V	mean	$4 * 10^{-9}$	$3.9 * 10^{-9}$	$4 * 10^{-9}$
	sigma	$2.3 * 10^{-9}$	$2.2 * 10^{-9}$	$2.3 * 10^{-9}$
@0.4V	mean	$1.76 * 10^{-9}$	$1.68 * 10^{-9}$	$1.79 * 10^{-9}$
	sigma	$1.4 * 10^{-9}$	$1.35 * 10^{-9}$	$1.48 * 10^{-9}$
@0.5V	mean	$8.2 * 10^{-10}$	$7.9 * 10^{-10}$	$8.3 * 10^{-10}$
	sigma	$4.9 * 10^{-10}$	$4.5 * 10^{-10}$	$5.1 * 10^{-10}$

Table 10: Write Energy

		Proposal	10T-1	10T-2
@0.3V	mean	$4.7 * 10^{-16}$	$4.6 * 10^{-16}$	$4.7 * 10^{-16}$
	sigma	$6.8 * 10^{-16}$	$6.9 * 10^{-16}$	$4.9 * 10^{-16}$
@0.4V	mean	$9.9 * 10^{-16}$	$9.4 * 10^{-16}$	$9.9 * 10^{-16}$
	sigma	$2.7 * 10^{-15}$	$2.7 * 10^{-15}$	$2.7 * 10^{-15}$
@0.5V	mean	$1.13 * 10^{-15}$	$1.04 * 10^{-15}$	$1.14 * 10^{-15}$
	sigma	$5.5 * 10^{-15}$	$5.5 * 10^{-15}$	$5.5 * 10^{-15}$

Technology Scaling

Figure 56 compares the read-0 delay at 45nm and 32nm. At 32nm technology, the proposed bitcell @0.3V is able to read 0 in 3.2ns(32bits/col) , 4.4ns(64bits/col) and 6.4ns(128bits/col) while only the 10T-Calhoun is able to read 0 in 8.8ns(32bits/col) and 15.6ns(128bits/col); thus @0.3V our proposal provides a speedup of 2.3x and 3.5x respectively. At 0.4V the speedup varies from 2x(32bits/column) to 3.8x(128bits/column). At 0.5V the speedup in access time is between 1.3x(32bits/column) and 2.2x(128bits/column) when reading a 0.

The energy comparison for 45nm and 32nm is shown in Figure 57 and Figure 58. The proposed bitcell reduces the read-0 operation total energy by at least - 40%(45nm) and 25%(32nm) @0.3V, 68%(45nm) and 67%(32nm) @0.4V, and 72%(32nm) and 70%(32nm) @0.5V.

Thus, with the proposed bitcell, columns with larger sizes can be used with comparatively smaller delays and energy expenditure, allowing the possibility of large size memories for ultra-low voltage operation.

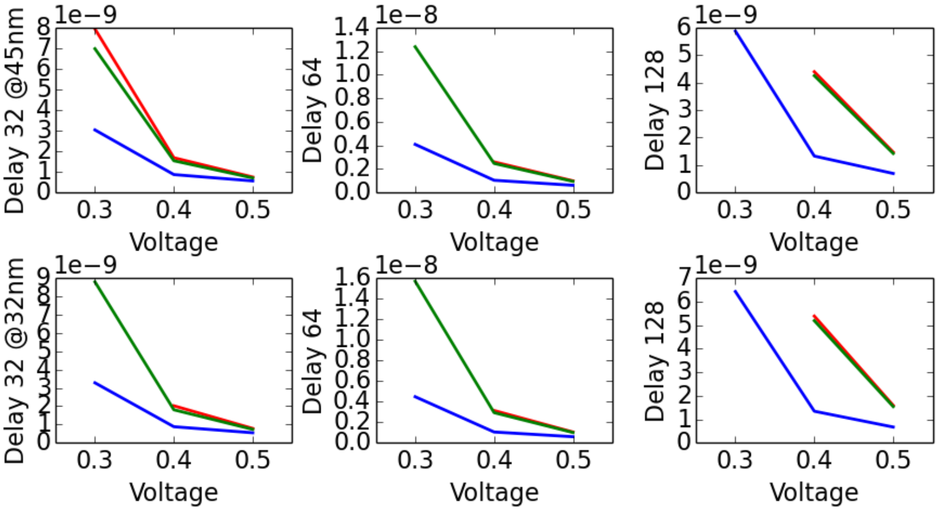


Figure 56: Read-0 delay comparison for a) 45nm b) 32nm technology

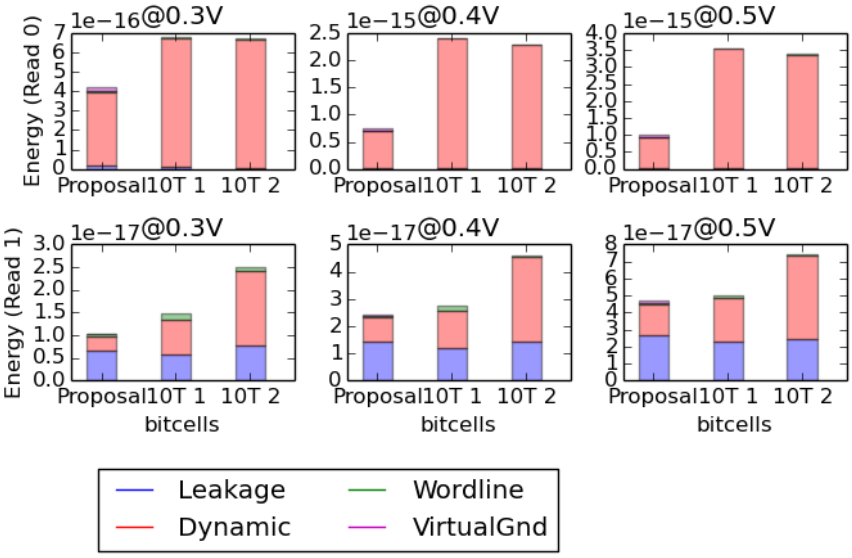


Figure 57: Energy comparison for Read '0' and Read '1' operation of bitcell @0.3V,0.4V and 0.5V supply voltage at 45nm

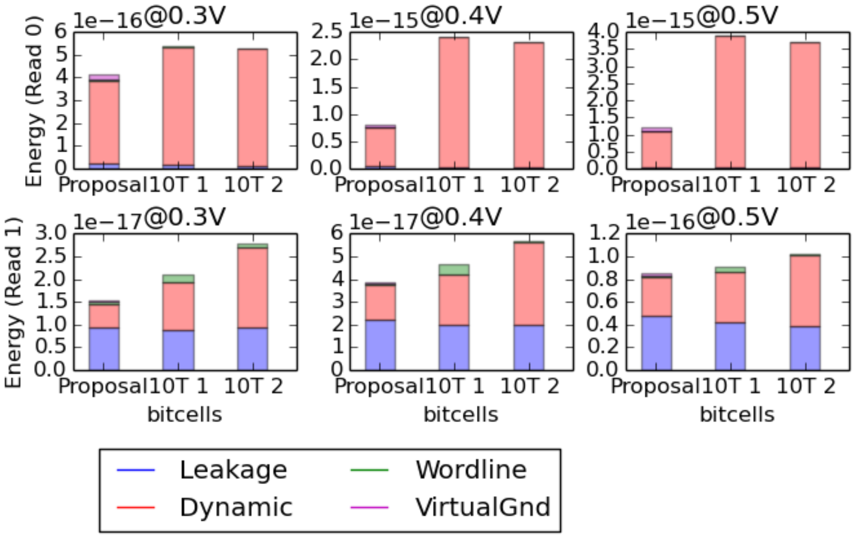


Figure 58: Energy comparison for Read '0' and Read '1' operation of bitcell @0.3V,0.4V and 0.5V supply voltage at 32nm

5.5 Conclusion

In this chapter, we presented a faster read access bitcell, 10TSD for subthreshold/near-threshold operation using virtual-ground to maintain high Ion/Ioff ratio. The bitcell outperforms traditional 10T bitcells in both energy and speed metrics at the cost of 13.6% larger area. The Read-0 access delay is reduced by a factor of 2x ~ 3x for column sizes between 128bits/col to 32bits/col. Also, the proposed 10T bitcell is the only 10T bitcell capable of reading 0 at the ultra-low voltage of 0.3V for short read-wordline pulse widths. The bitcell has a smaller number of 'on' transistors during the read operation thereby lowering the dynamic energy consumption by 70% and 30% (when reading a 0 and a 1, respectively) at the cost of a slightly higher current.

Hacker: The statistics are irrefutable...
Humphrey: Statistics? You can prove anything with statistics.
Hacker: Even the truth.
Humphrey: Yes... No!

Yes, Prime Minister: The Smoke Screen

6 2T and 3T1D e-DRAM gain cells as alternative to SRAM for Minimum Energy Operation

6.1 Introduction

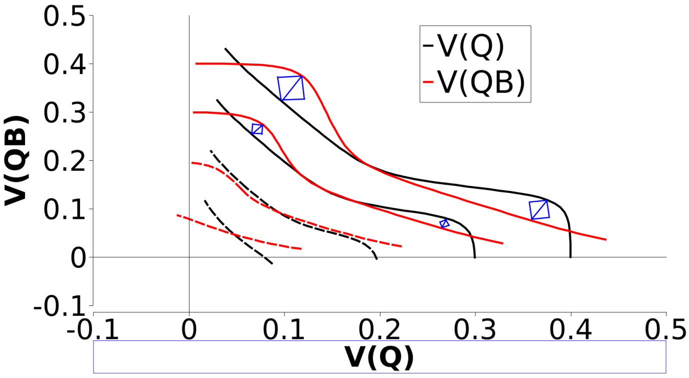
The emergence of Internet-of-Things (IOT) has opened up new opportunities to collect data for analysis in the cloud using wireless battery-operated wearable sensors. The number of these devices is expected to increase to 35 sextillion units in 2020 [Manyika et al., 2011] finding use cases in many domains which were till yet silicon-free. Achieving a smaller form factor and higher energy-efficiency is of prime importance in a bio-medical wearable devices. Recently, embedded-DRAM (e-DRAM) caches have been advocated as the successors of SRAM [Amat et al., 2014, Chun et al., 2009, Chun et al., 2011, Iqbal et al., 2012, Liang et al., 2008] considering their higher densities (> 2X)[Teman et al., 2012] and smaller leakage, due to fewer number of transistor. 3T1D e-DRAM gain-cell is shown to be capable of achieving access speeds comparable to 6T SRAM [Liang et al., 2008] and with larger device density [Chun et al., 2009]. The maximum energy efficiency has been shown to exist at sub-threshold circuit operation [Calhoun et al., 2005, Hanson et al., 2006]. However the 6-Transistor SRAM bit-cell cannot provide enough reliability because of its reduced noise margin at these ultra-low voltages. Operating e-DRAMs at sub-threshold/near-threshold region offers the next step in the direction of increasing energy-efficiency of wearable bio-medical health-monitoring systems.

6.2 Background

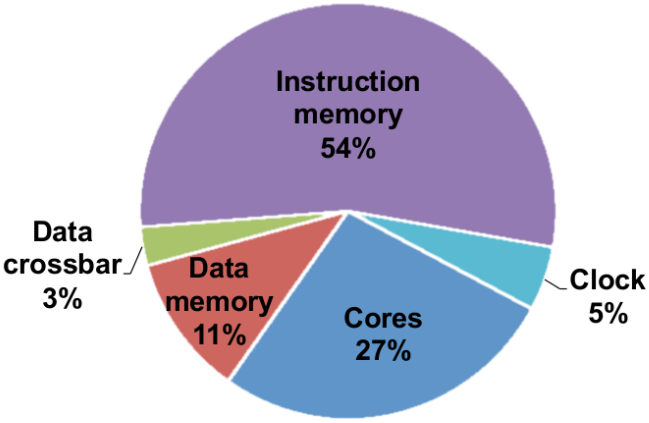
The energy consumption in CMOS circuits is mainly constituted of the dynamic energy and leakage energy. The former is spent in switching capacitive loads and the later is consumed by sub-threshold leakage currents when the transistors are off. Dynamic energy of the circuit can be decreased quadratically by scaling supply voltage (V_{DD}). When the V_{DD} is aggressively scaled down to sub-threshold voltages, the driving-current (I_{on}, $V_{GS} = V_{DD}$) and the off-current (I_{off}, $V_{GS} = 0$) are given by the equation,

$$I_{SUB} = I_o e^{V_{GS} - V_{th}/nV_T}$$

The delay (t_d) of the circuit increases exponentially when the supply voltage is scaled to sub-threshold region thereby increasing the leakage energy per operation of the circuit. The MEP of the circuit can be achieved at V_{DD} in the sub-threshold region[Calhoun et al., 2005, Hanson et al., 2006]. However, the operating voltages for a processor are limited to the minimum-voltage required for the reliable operation of on-chip SRAM cache which fails when scaling down to ultra-low voltages because of its shrinking noise margins, Figure 59a. Nevertheless, SRAM dominates the energy consumption among the components of a processor [Dogan et al., 2012], (Figure 59b) and several alternative SRAM bit-cells have been proposed. These sub-threshold SRAM bit-cells have 8-transistors[Verma and Chandrakasan, 2008], 10-transistors [Kulkarni et al., 2007, Calhoun and Chandrakasan, 2007, Chang et al., 2009] (proposed bitcell in previous chapter) or more.



(a) The voltages in cross-coupled latches (Q and QB) of minimum feature-size 6T SRAM ($\beta = 1$) are plotted against one another giving read butterfly curve. Read Noise margin is the length of the largest embedded square in-between the two lobes of the curve. Noise margin vanishes below 0.3V supply voltage.



(b) Power distribution in a multi-core architecture for bio-medical applications, source [Dogan et al., 2012]. Memory is the highest power consuming component.

Figure 59

As an alternative to SRAM bit-cells, [Meinerzhagen et al., 2013] investigated sub-threshold 2T e-DRAM gain-cells for ultra-low power medical applications. Their study showed reliable operation for 2kb e-DRAM array up to sub-threshold voltage of 0.4V at mature 0.18μm node and up to near-threshold voltage of 0.6V at scaled 40nm node. Further, [Amat et al., 2014] observed that the 3T1D gain-cells exhibits better reliability in front of device variability and single event upsets than the 2T gain cell.

6.2.1 2T and 3T1D gain cells

2T and 3T1D gain cells are two-port memories with separate read and write paths as shown in Figure 60, which also shows the wave-forms for their read/write operation. Since the leakage current of the NMOS transistor is significantly higher than that of the PMOS transistor, alternate cell configurations that mix the transistor types (PMOS write transistor and NMOS transistors for the read path) achieve better memory cell performance than the NMOS-only gain cell design [Amat et al., 2014, Chun et al., 2009, Amat et al., 2015]. The storage node capacitor (SN), formed by T2's gate capacitance and T1's diffusion capacitance, stores the data as charge. To write data into the gain cell, T1 is turned on to transfer charge from BL_{Write} to SN. Figure 61 shows the MEP for read operation of 3T1D gain-cell and 6T SRAM bitcell. The 6T bitcell fails to hold value during read operation below 0.3V, as seen in Figure 59a, and it has read MEP energy roughly 200X that of 3T1D.

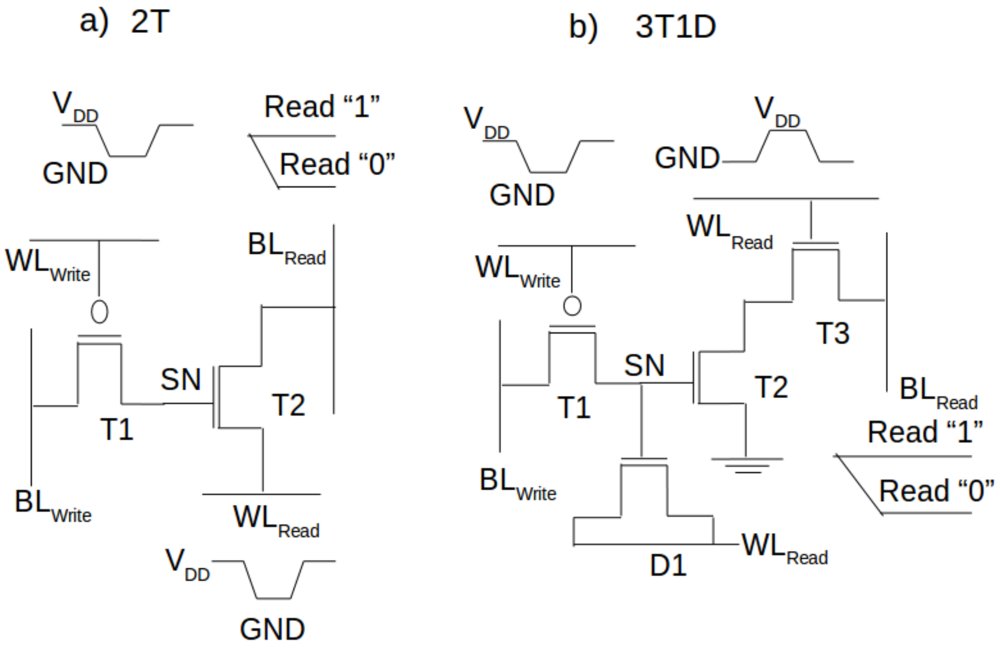


Figure 60: Schematic of (a) 2T and (b) 3T1D gain cell. Read operation begins by pre-charging the read bit-line. Subsequently read word-line is driven low for 2T and high for 3T1D gain cell to complete the read operation.

6.3 Methodology

We study the energy-efficiency of 2T and 3T1D e-DRAM gain-cells within the following design space:

1. Different sizing of transistors: Nominal transistor sizes are taken from [Lovin et al., 2009]. The lengths and widths are increased in the range [1x, 2x, 3x, 4x] for each one of the e-DRAM cell transistors.

2. Wordline assist: A voltage offset in the range [0 to 0.2V] is applied to WL_{Read} and WL_{Write}.

Δrwl During a read operation, over-drive the WL_{Read} for the 3T1D and under-drive the WL_{Read} for the 2T. The effect is a faster read access and reduction in the read leakage energy. During standby (retention), under-drive the WL_{Read} for the 3T1D and over-drive the WL_{Read} for the 2T. The effect is a decrease in sub-threshold leakage through the read path.

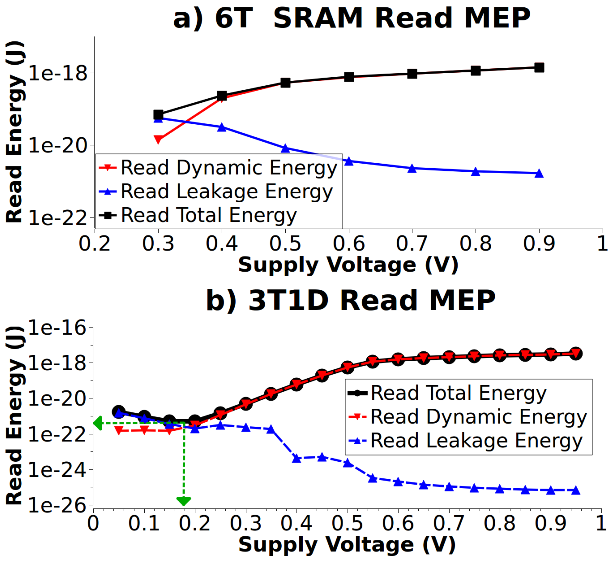


Figure 61: Read minimum energy point (MEP) for a) 6T SRAM is $7 * 10^{-20}$J at 0.3V. b) 3T1D gain-cell is $4 * 10^{-22}$J at 0.2V.

Δwwl During read and standby (retention), over-drive WL_{Write} to decrease the sub-threshold leakage through the write path.

3. High threshold voltage transistors: High threshold voltage transistors with ΔVth in the range [0 to 0.2V].

4. Temperature: The operating temperature is varied from $-70°C$ to $100°C$

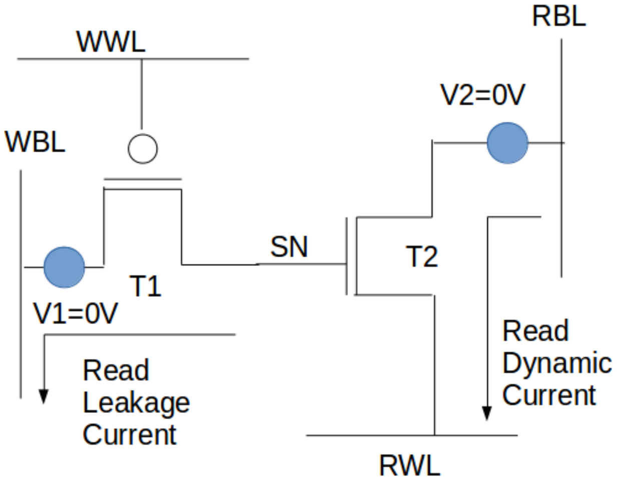


Figure 62: The zero-voltage sources V1 and V2 are added to the write and read path. The current through these voltage sources is measured to estimate the leakage and dynamic energies during the read operation.

These e-DRAM gain-cell designs are compared under the following metrics:

- *Minimum-energy point (MEP):* The dynamic and leakage energies of the gain-cell are estimated by measuring current flowing through the zero-voltage sources, V1 and V2, in the read and write path as shown in Figure62 with 2T gain-cell as an example. The MEP read energy is defined as the sum of Read-0 and Read-1 energy at MEP voltage. The voltage sweep required to estimate MEP is performed down to 0.1V.

- *Access Delay at MEP:* The read delay is measured as the time from the instant the read wordline is activated till the read bitline voltage decreases by 0.03V, assuming sense amplifier can sense 30mV input voltage difference [Wicht et al., 2004].
- *Retention Time (RT) at MEP:* In this work, we measured it as the time it takes for the stored logic at SN to deteriorate till half of the supply voltage. This is different from its definition for above-threshold operation, where it is defined in terms of the threshold voltage of the read transistor T2 - Retention "0" (or "1") as the time it takes for V_{SN} to rise (or fall) to $V_{th,T2}$. Since, the operating voltages for our analysis are in sub-threshold region, we forgo the above-threshold definition and instead consider half-V_{DD} as the limit for V_{SN} in both Retention-"0" and "1" cases.

The spice net-lists of the 2T and 3T1D gain-cells are simulated in HSPICE [HSP,] circuit simulator. The e-DRAMs were shown to perform reliably in near-threshold region at 40nm node in [Meinerzhagen et al., 2013]. So in this work, e-DRAM gain-cells are studied at the next scaled technology node 32nm (using HP PTM models [PTM,]) which is going to be the technology node for the future sub-threshold circuit implementations.

6.3.1 Kriging Meta-model for nominal(without-variation) case

In the design-space with four levels per parameter, there exists 262,144 (4^9) designs for a 2T cell (2 lengths, 2 widths, 2 High Vth transistors, read and write wordline boosting and a temperature parameter) and 1,073,741,824 (4^{15}) designs for a 3T1D cell. Furthermore, a voltage sweep needs to be performed at each of these design points to estimate the MEP. Design exploration with these many simulations can be very time expensive.

Hence, we first build meta-models (i.e, surrogate model) to predict the MEP read energy, MEP read delay and MEP retention time with the transistor dimensions, wordline boosting voltages, threshold voltages of transistors and temperature as parameters of the model. The predictions from the model are used to find the optimal regions in the parameter space and these regions are then simulated in HSPICE to get accurate results.

Here, we use the statistical regression method known as Kriging. In polynomial regression, we model the simulation output, $y(x)$, with $x = (x_1, x_2, ..., x_n)$ as input variables (factors) by the following equation,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + e \tag{36}$$

where y is the observed value, β_i are the parameters of the model and e is the residual error which includes both the intrinsic noise and the lack-of-fit of the model. The error term e is assumed to be independent and normally distributed (iid assumption). The linear regression can thus be seen as the modeling of the $y(x)$ by a deterministic function with β_i parameters and a random process e with constant term diagonal matrix as its covariance (because of the iid assumption).

Kriging is a generalization of this approach, it models the simulation output as the sum of a deterministic trend function $\mu(x)$ and a random process $R(x)$ where kernels such as linear, periodic or RBF can be used as the covariance function of the random process $R(x)$ to capture any spatial relations in the simulation output, $y(x)$.

$$y_{kriging}(x) = \mu(x) + R(X) \tag{37}$$

When the deterministic function $\mu(x)$ is assumed to be a constant (when it is zero, the kriging model is said to be centered) and the random process $R(x)$ is assumed to be the Gaussian random process, then the kriging meta-model is the same as the Gaussian Process regression

technique that we had used earlier in Chapter 4. The main advantage of kriging (and hence of Gaussian process regression) is that it interpolates the simulated data with the zero model error at the simulated points. We use a universal kernel (Matern kernel) as the covariance function of the random Gaussian process $R(x)$ for interpolation. This kernel can interpolate functions that are less smooth (have finite derivatives) than those obtained from the RBF kernel (have infinite derivatives). In contrast, we had used additive kernels in Chapter 4 because there we wanted to increase the extrapolation capability of the model.
To create these meta-models, 1000 points are sampled using the Latin-Hypercube-Sampling (LHS) method to produce a space-filling design. A space filling design has two objectives:

1. Maximize the minimum distance between any two design points

2. Spread the design point uniformly in the design space

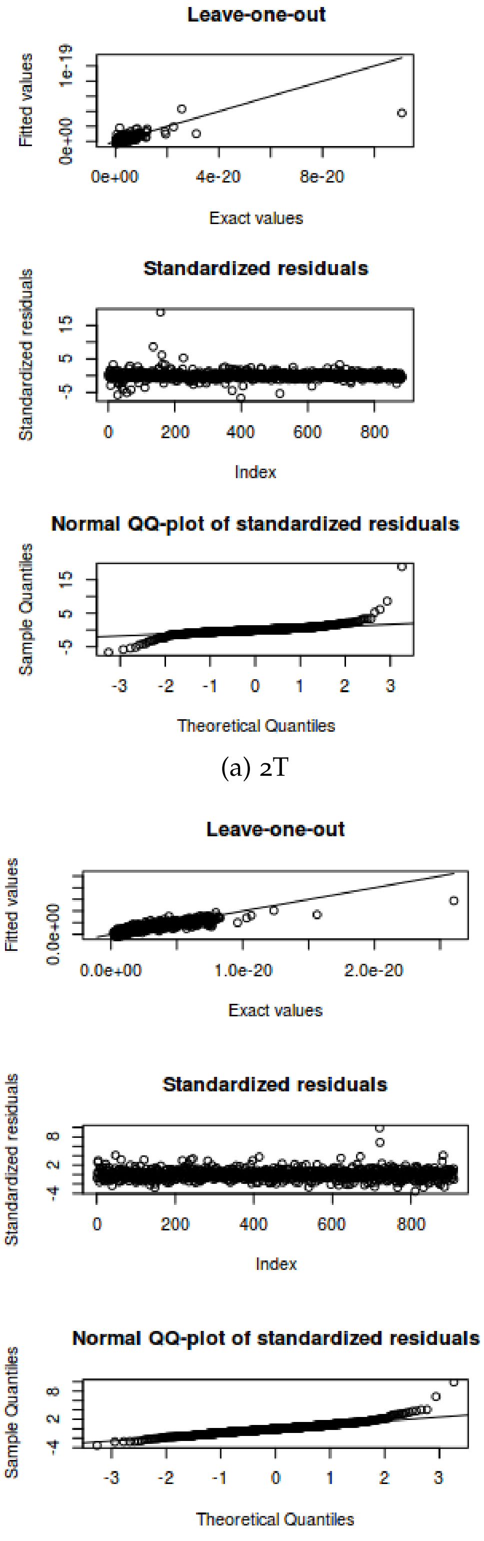
Latin-Hypercube-Sampling (LHS) is a space-filling design which additionally ensures that the design points are also spread uniformly over the range of each input variable. However, for a high-dimensional space, the distribution of points provided by LHS may deviate considerably from a uniform distribution (leading to high-discrepancy). Thus, an additional step of LHS optimization is performed, using the Enhanced Stochastic Evolutionary (ESE) algorithm provided in the DiceDesign package of R [Dupuy et al., 2015]. The model is cross validated by leave-one-out which gives coefficient of determination (R^2) 0.73 for 2T MEP energy and 0.69 for 3T1D gain cell. The validation plots for the regression model of the 2T and 3T1D gain cells are shown in Figure 63.

6.3.2 Full Factorial Analysis in the presence of process variations

In the presence of random process variations, it is necessary to separate the effect of up-sizing the transistor dimensions on the MEP read energy from that of the random aspect of process variations. The mean value of the MEP read energy estimated using Monte Carlo method will vary between the separate runs of the method. This will also be true for the Monte Carlo estimate of the change in the MEP read energy for different up-sized designs compared to the nominal sized design. We need to be sure that the difference we observe is because of the effect of the design variable and not because of the randomness in the process variations. So, to effectively compare the up-sized designs, the confidence intervals for their improvement in MEP read energy are needed. For this, a 2^k full factorial design experiment with 5000 replications is done for up-sized designs (lengths and widths of transistors with two levels [1x, 4x]). The variability in threshold voltage is assumed to be 6% following the EU project statement [Rubio Sola et al., 2012]. A full factorial design experiment means that all combinations of the different levels of design variables are simulated. This allows us to identify the significant design variables and also study their interactions. The decision on whether the difference in the mean MEP energy observed is due to the effect of the design parameters or due to the randomness in the process parameters is made by calculating the p-values from ANOVA test [Box et al., 1978].

ANOVA test

ANOVA test is related to the class of regression techniques. In standard regression, we model the effect of “continuous" predictors (independent variables) on a “continuous" response (dependent variable). Similarly, in logistic regression we model the effect of “continuous" predictors on a "categorical" response which has two or more levels, such as whether the memory read operation fails or not. ANOVA modeling is used when we have “categorical" predictors and “continuous" response. In our study, the transistor dimensions are categorical predictors with two levels [1x, 4x] and the response of mean MEP read energy is a continuous variable.



(a) 2T

(b) 3T1D

Figure 63: Regression validation for kriging model of 2T and 3T1D MEP energy. The top plot shows the agreement between predicted and actual values using leave-one-out cross validation. The middle plot verifies the assumption that residuals are randomly distributed around zero without any drift. The bottom plot verifies the assumption that residuals are almost normally distributed.

Here, we illustrate the basic idea behind ANOVA for a one-way test (that is, when we are studying the effect of only factor with multiple levels, such as read transistor width with 1x and 4x up-sizing levels). The objective is to partition the total variation observed in the MEP energy from all the samples in the design experiment, between two parts:

1. *Effect-Variance or "Between group variance":*
   Variation in MEP energy that cannot be explained just by the randomness in the threshold voltage variations and hence can be ascribed to the up-sizing of the read transistor width. To estimate its value, first we calculate SS_{group} which is the sum of the squared deviations of mean MEP energy (M_{1x} and M_{4x}) at the up-sizing levels 1x and 4x, from the global mean μ. Effect-Variance is then estimated by dividing SS_{group} by its degrees of freedom ν_1. ("n" is the number of random samples simulated in each up-sizing level).

$$\text{Effect-Variance} = \frac{n(M_{1x} - \mu)^2 + n(M_{4x} - \mu)^2}{\nu_1} \tag{38}$$

2. *Error-Variance or "Within group Variance":*
   Variation in MEP energy that can be explained by the randomness in threshold voltage variations. To estimate its value, we first calculate SS_{error} which is the sum of squared residual errors at up-sizing levels 1x and 4x. That is, the sum of the squared deviations of 1x up-sized samples $y_{1x,i}$ from their mean M_{1x} and the squared deviations of 4x up-sized samples $y_{4x,j}$ from their mean M_{4x}. An indirect method to estimate SS_{error} is to estimate SS_{total} which is the sum of squared deviations from the global mean of all samples and take the difference $SS_{total} - S_{group}$. Error-Variance is then estimated by dividing SS_{error} by its degrees of freedom ν_2.

$$\text{Error-Variance} = \frac{\sum_i^n (y_{1x,i} - M_{1x})^2 + \sum_j^n (y_{4x,j} - M_{4x})^2}{\nu_2} \tag{39}$$

Degrees of freedom

The degrees of freedom of a variance is an estimate of the amount of independent information ("free variables") available to estimate the variance. In our example case of one-way test, assume both factor levels have 5000 samples. Thus, we have N=10,000 total samples, from which we estimate the global mean by taking the average value of the samples' MEP energies. Since we estimated the SS_{total} , SS_{group} and the SS_{error} by assuming that the global mean is known and fixed; we can only modify $10,000 - 1 = 9,999$ samples so that the global mean estimate does not change. Furthermore, note that the global mean is also the weighted average of the group means (weights are equal in our case because each factor level has same number of samples, i.e 5000). So we can only modify $2 - 1 = 1$ group mean to keep the global mean estimate fixed. Thus, we have the following situation (taking N=10,000 and K=2):

1. We estimated SS_{total} over $N - 1$ "free variables" (degrees of freedom). Hence, the mean of SS_{total} (**Total-Variance**) is estimated (without bias) by dividing SS_{total} by $N - 1$.

2. We estimated S_{group} over $K - 1$ degrees of freedom. Hence the mean of SS_{group} (**Effect-Variance**) is estimated by dividing SS_{group} by $K - 1$.

3. The **Error-Variance** is left with the remaining $(N-1) - (k-1) = N - K$ degrees of freedom. So it is estimated by dividing SS_{error} by $N - K$.

Hypothesis Testing

Error-Variance is considered as the *noise* in our data and the Effect-Variance as the *signal* present in the data. The more noise is present in the data, the more difficult it is to find the signal. If the Error-Variance is significantly larger than the Effect-Variance, then the observed difference in MEP energy between factor levels could be because of the random variations.

We quantify the difference in magnitude between the two variances by taking their ratio,

$$\text{F-value} = \frac{\text{Effect-Variance}}{\text{Error-Variance}} \tag{40}$$

This signal-to-noise ratio is called as F-value and we want to be able to test the following two hypothesis using this F-value:

1. **Null Hypothesis:** There is no effect of up-sizing read transistor width on the mean MEP energy. That is, the 5000 read MEP energy random samples for 1x up-sized design and 4x up-sized design are from the same MEP energy distribution, $\mu_{1x} = \mu_{4x} = \mu$.

2. **Alternate Hypothesis:** The mean MEP energy changes by up-sizing read transistor width. That is, the distribution of MEP energy of the 4x up-sized design is different from the distribution of the 1x up-sized read width design, $\mu_{1x} \neq \mu_{4x}$.

The statistical testing of the two hypothesis can be done in the following steps:

1. We find the *distribution of the F-value under null hypothesis*. That is, we find the probability distribution of F-values by estimating the Effect-Variance and Error-Variance using the *same MEP energy distribution* for both the up-sizing levels.

2. We estimate the *F-value from the simulation results* of 1x up-sized design and 4x up-sized design, each with 5000 random samples.

3. We reject the **Null Hypothesis** only if the our estimate of F-value in the previous step has very small probability (*p-value*) of occurring in the F-value distribution under null hypothesis. The probability needed to reject the null hypothesis is called **significance level**, α. If we take $\alpha = 0.001$, then we will have one in a thousand chance of wrongly rejecting the null hypothesis, that is, attributing the change in mean MEP energy to the read transistor width up-sizing when actually there is no difference between the two MEP energy distributions.

Finding the F-value distribution under Null Hypothesis

The Effect-Variance and Error-Variance were calculated using the sum-of-squared deviations, SS_{error} and SS_{group}. If the random samples (that is, the MEP energy values) used to estimate SS_{error} and SS_{group} are independent and *normally distributed*, $N(\mu_{1x}, \sigma^2_{1x})$ and $N(\mu_{4x}, \sigma^2_{4x})$ [**ANOVA Assumption: Normality and Independence**] and further $\sigma^2_{1x} = \sigma^2_{4x} = \sigma^2$ [**ANOVA Assumption: Homoscedasticity**]. Then, using these sum-of-squared deviations we can get the following chi-squared random variables:

$$\begin{aligned} \chi_{1x,i} &= \frac{(y_{1x,,i} - \mu_{1x})^2}{\sigma^2} \\ \chi_{4x,j} &= \frac{(y_{4x,j} - \mu_{4x})^2}{\sigma^2} \end{aligned} \tag{41}$$

A property of chi-squared random variables is that the sum of independent chi-squared random variables is also a chi-squared random variable. Thus we get the following chi-squared random variables,

$$\begin{aligned} \chi_{1x} &= \sum_{i}^{n} \frac{(y_{1x,,i} - \mu_{1x})^2}{\sigma^2} \\ \chi_{4x} &= \sum_{j}^{n} \frac{(y_{4x,j} - \mu_{4x})^2}{\sigma^2} \end{aligned} \tag{42}$$

In calculation of Effect-Variance, we had to estimate the means, M_{1x} and M_{4x} at 1x up-sizing and 4x up-sizing. These two estimates are also random variables and are normally distributed, $M_{1x} \sim N(\mu_{1x}, \sigma^2/n)$ and $M_{4x} \sim N(\mu_{4x}, \sigma^2/n)$. So $\sqrt{n}M_{1x} \sim N(\sqrt{n}\mu_{1x}, \sigma^2)$ and $\sqrt{n}M_{4x} \sim N(\sqrt{n}\mu_{4x}, \sigma^2)$. This gives us following chi-squared random variables

$$\begin{aligned} \chi_{M_{1x}} &= \frac{(\sqrt{n}M_{1x} - \sqrt{n}\mu_{1x})^2}{\sigma^2} = \frac{n(M_{1x} - \mu_{1x})^2}{\sigma^2} \\ \chi_{M_{4x}} &= \frac{(\sqrt{n}M_{4x} - \sqrt{n}\mu_{4x})^2}{\sigma^2} = \frac{n(M_{4x} - \mu_{4x})^2}{\sigma^2} \end{aligned} \tag{43}$$

When the **Null Hypothesis** is true (that is $\mu_{1x} = \mu_{4x} = \mu$), then all the random samples y_i in the experiment are from the same normal distribution, $N(\mu, \sigma)$:

$$\begin{aligned} \chi_{SS_{error}} &= \frac{SS_{error}}{\sigma^2} = \chi_{1x} + \chi_{4x} \\ \chi_{SS_{group}} &= \frac{SS_{group}}{\sigma^2} = \chi_{M_{1x}} + \chi_{M_{4x}} \\ z_i &= \frac{(y_i - \mu)}{\sigma} \text{ is a standard normal variable} \\ \chi_{SS_{total}} &= \frac{SS_{total}}{\sigma^2} = \frac{\sum_{i}^{N=2n}(y_i - \mu)^2}{\sigma^2} = \sum_{i}^{N} {z_i}^2 \end{aligned} \tag{44}$$

Thus we have, under the **Null Hypothesis**:

$$\chi_{SS_{total}} = \chi_{SS_{group}} + \chi_{SS_{error}} \tag{45}$$

Cochran's theorem states that this is only possible if $\chi_{SS_{group}}$ and $\chi_{SS_{error}}$ are independent random variables. If we take the ratio of two *independent* chi-squared random variables scaled by their degrees of freedom, we get Fisher-distribution (also called F-distribution). Thus if **Null Hypothesis** is true, then F-values are distributed according to the F-distribution:

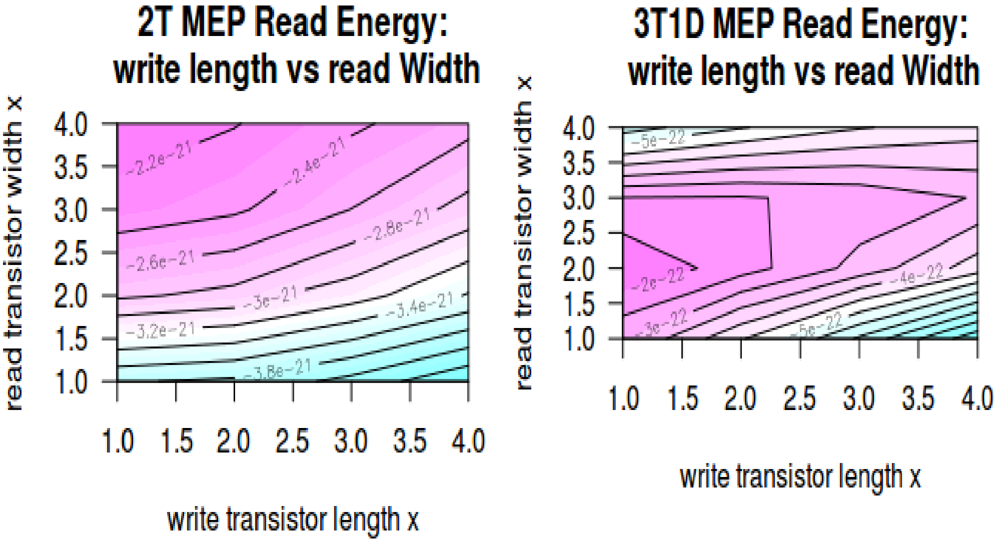
$$\text{F-value} = \frac{SS_{group}/((K-1)\sigma^2)}{SS_{error}/((N-K)\sigma^2)} = \frac{\chi_{SS_{group}}/(K-1)}{\chi_{SS_{error}}/(N-K)} \sim F_{K-1,N-K} \tag{46}$$

We identify statistically significant design parameters by using the significance level of 0.001.

6.4 Results

6.4.1 Nominal Analysis (without process variations)

6.4.1.1 Sizing



(a) 2T

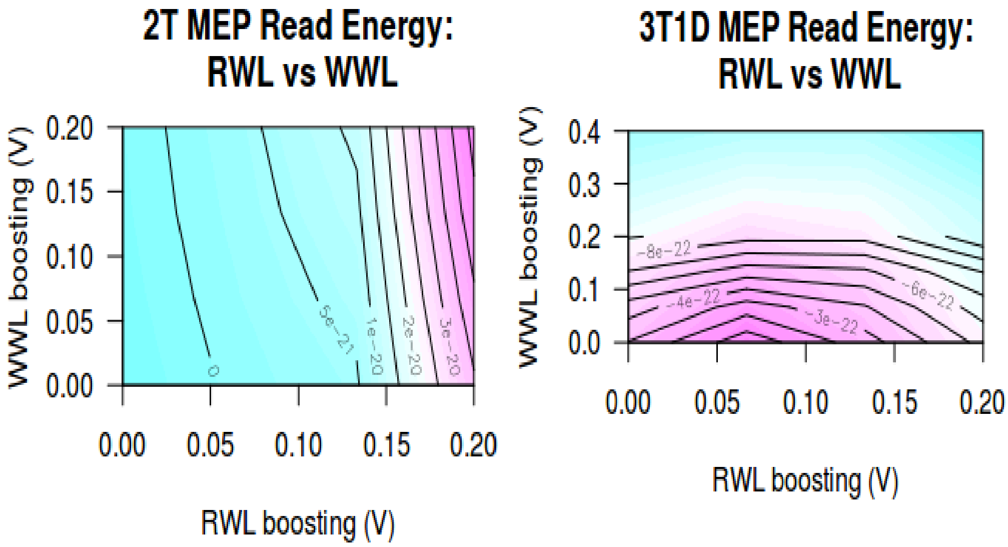
(b) 3T1D

Figure 64: Contour plots for MEP energy when up-sizing transistors. Increasing the write transistor length decreases MEP energy while increasing read transistor width increases MEP energy. Increasing both together keeps the MEP energy same. Color-map is blue for low MEP energy and pink for high MEP energy

The width of the read transistor is typically up-sized to increase the retention time. This however increases the MEP energy. The contour plot in Figure 64 shows that it is possible to decrease MEP energy when up-sizing the write transistor length while also up-sizing the read transistor width. The HSPICE simulation of 4x write transistor length design shows a decrease in MEP energy by 29% for 2T and 26% for 3T1D.

6.4.1.2 Wordline Boosting

Applying read wordline boosting increases the MEP energy In contrast, the effect of write wordline boosting is to reduce the MEP energy. This can be seen in Figure 65. HSPICE simulations of 0.2V read wordline boosting design shows MEP energy is higher by 564% for 2T and 61% for 3T1D . While HSPICE simulation of 0.2V write wordline boosting design shows MEP energy is lower by 34% for 2T and 41% for 3T1D.



(a) 2T

(b) 3T1D

Figure 65: Contour plots for MEP energy when wordline boosting in applied. Boosting read word line (RWL) is increasing MEP energy. Boosting write word line (WWL) is decreasing MEP energy. Color-map is blue for low MEP energy and pink for high MEP energy

6.4.1.3 High Threshold Voltage Transistors

Using high threshold voltage transistors in the read and write paths to decrease leakage current has opposite effects on the MEP energy. While using high threshold transistors on the write path is reducing MEP energy, using high threshold transistors in the read path increases the MEP energy. This effect can be explained by the increase in the read delay which would consequently increase the read leakage energy. The contour plots in Figure 66 suggest that designs with high threshold transistors on both read and write path have lower MEP energy than designs with only high threshold read transistors. The HSPICE simulation of 0.2V higher threshold voltage for write transistor shows a decrease in MEP energy by 35% for 2T and 25% for 3T1D. The HSPICE simulation of the design with 0.2V higher threshold voltage read transistors shows an increase in the MEP energy by 860% for 2T and 293% for 3T1D.

6.4.1.4 Temperature

Increase in temperature increases the read MEP energy. However, the increase in energy can be reduced by also increasing the write length as in seen in Figure 67. HSPICE simulations show that at 100°C the increase in MEP energy is 116.9% for 2T and 130% for 3T1D. This increase is then reduced with the 4x up-sizing of write transistor length to only 12% for 2T and 23% for 3T1D.

In summary, the read MEP energy is reduced by either write wordline boosting or using write transistor with high threshold voltage or by up-sizing write transistor length for both 2T and 3T1D gain cells. Thus, reducing leakage current through write path is necessary to reduce MEP energy, especially at higher temperatures. On the contrary, reducing read delay by either up-sizing read transistor width or read wordline boosting increases the read MEP energy.

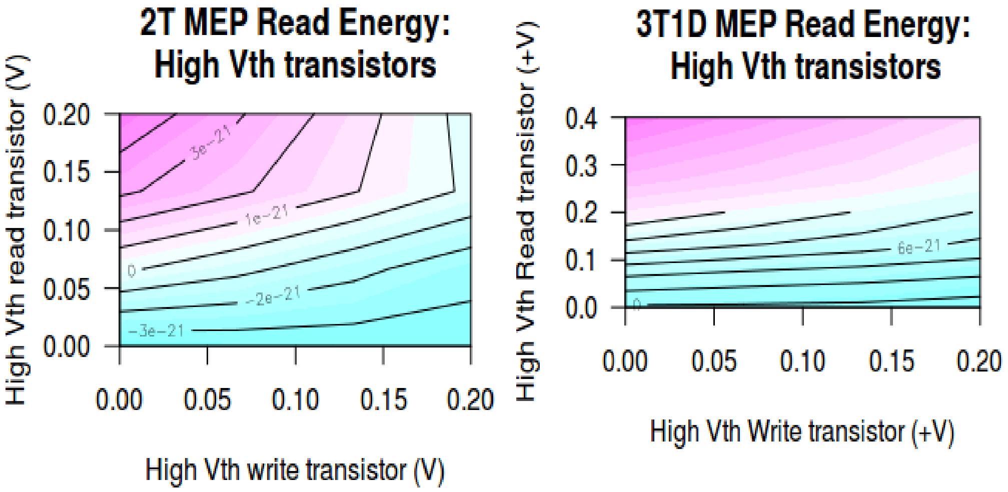
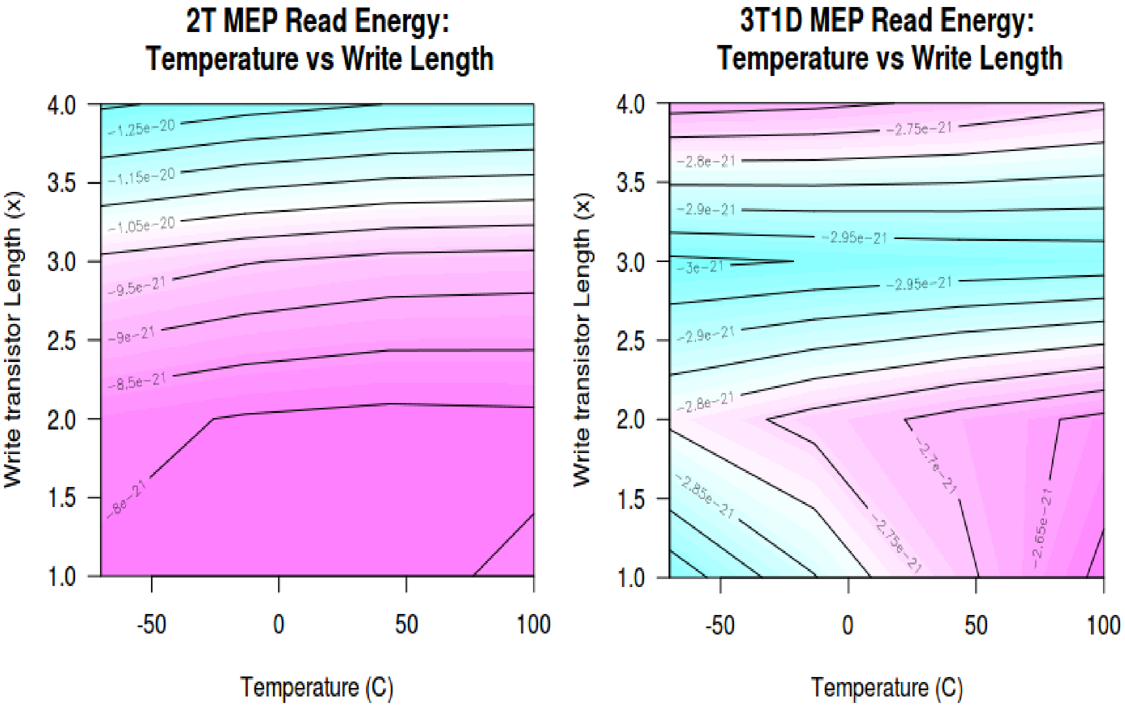


Figure 66: Contour plots for MEP energy when high threshold transistors are used, with x-axis and y-axis as ΔVth. A high threshold voltage transistor in the write path decreases MEP energy. In contrast, using a high threshold transistor in the read path increases the MEP energy. Color-map is blue for low MEP energy and pink for high MEP energy



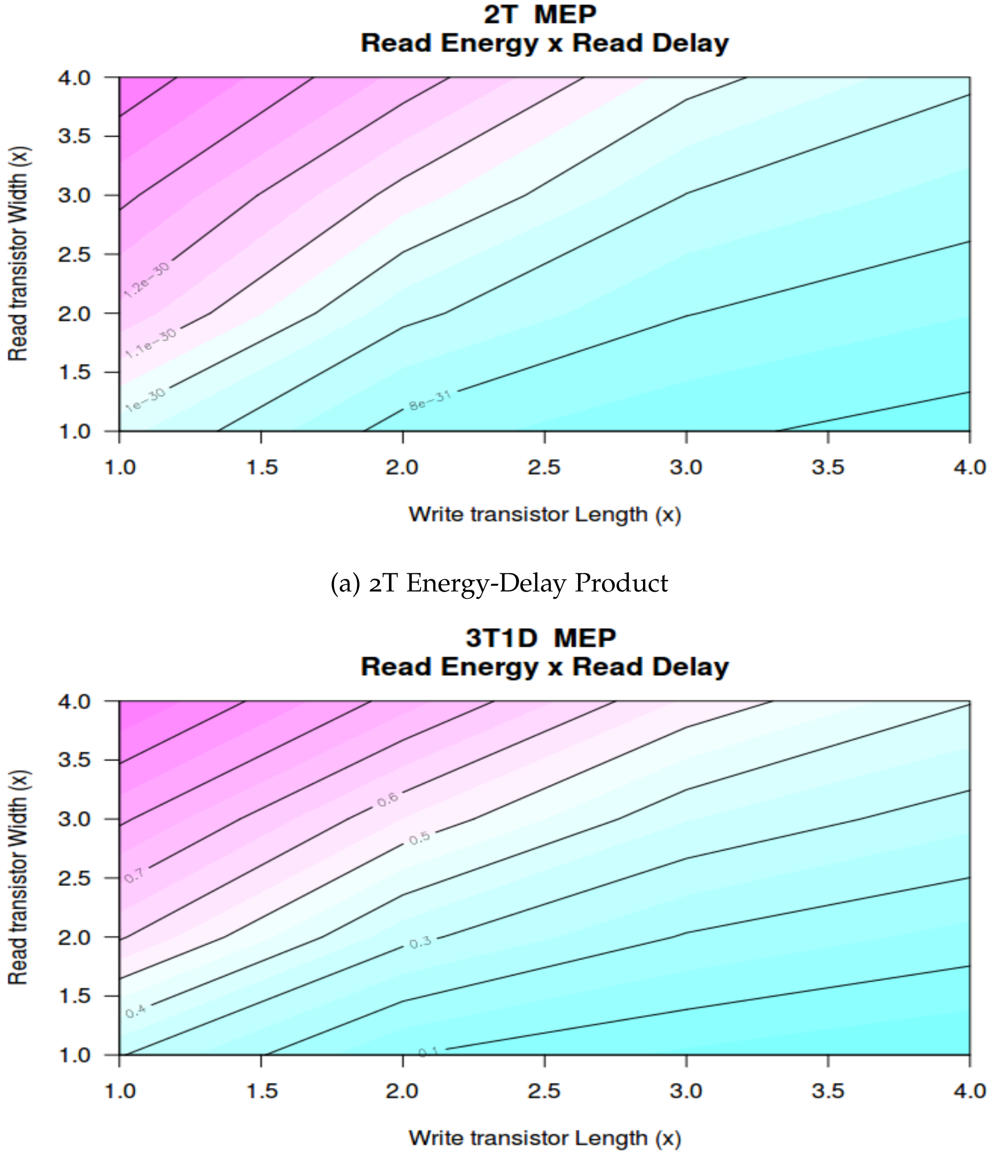
(a) 2T

(b) 3T1D

Figure 67: Temperature increases MEP energy. This can be mitigated by increasing the write transistor length. Color-map is blue for low MEP energy and pink for high MEP energy

6.4.2 Joint Optimization of read energy with read delay, Retention time

The designs with a smaller Read MEP energy and also smaller read delay are found by considering designs with least energy-delay product. The contour plot for this product is shown in Figure 68, which shows that up-sizing the write transistor length decreases the energy-delay product. In contrast, up-sizing the read-transistor width increases the energy-delay product. The HSPICE simulation of 4x write transistor length design shows that the energy-delay product is reduced by 30% for 2T and 26.3% for 3T1D.

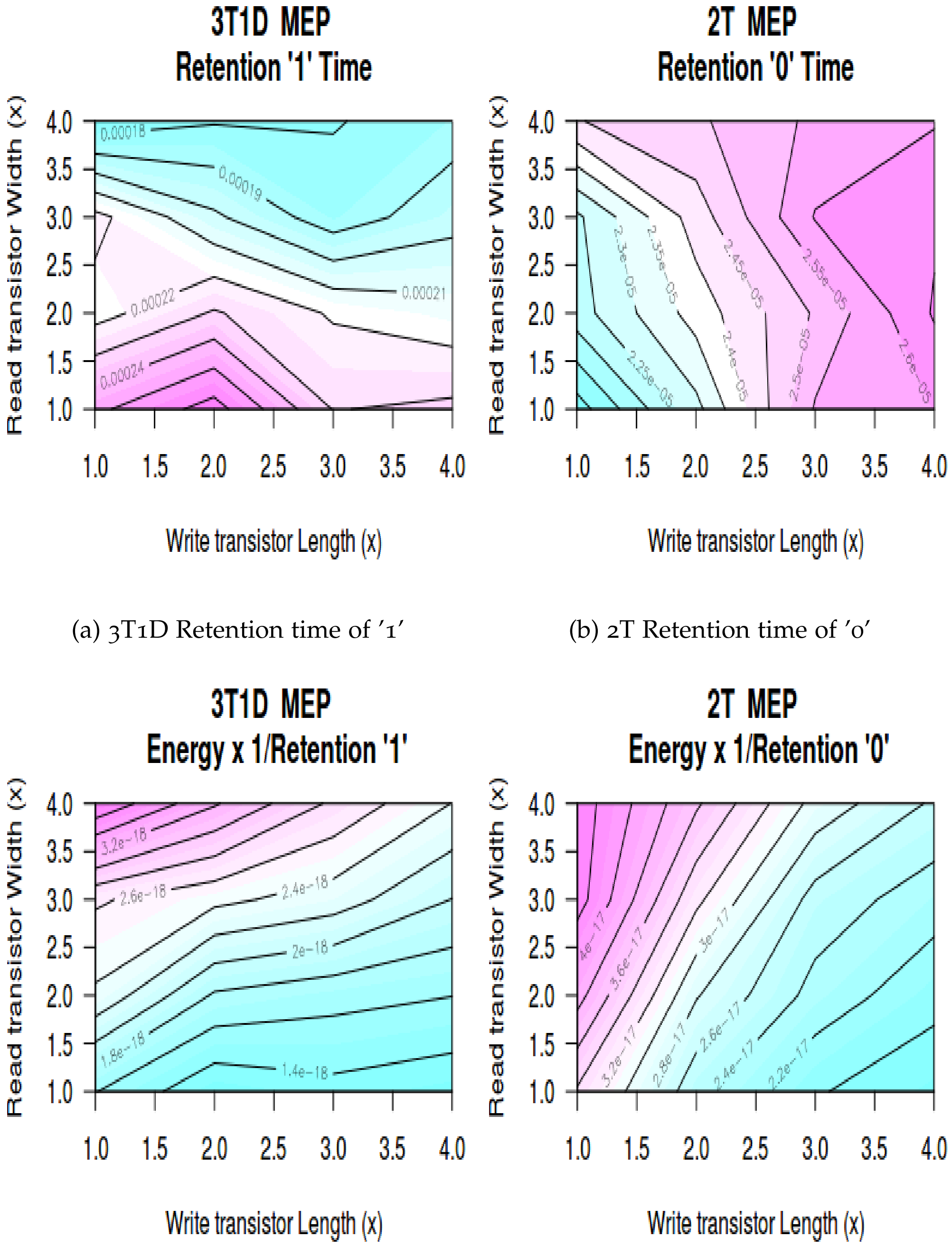


(a) 2T Energy-Delay Product

(b) 3T1D Energy-Delay Product

Figure 68: Contour plots for product of Read MEP energy and read delay. The design with smallest product would be the optimum point with less MEP energy and smaller read delay. Color-map is blue for low values and pink for high values

The HSPICE simulations showed that the retention time of 2T for stored value of ′1′ and of 3T1D for stored value of ′0′ is greater than 1ms for all up-sizing design options. The contour plots showing retention time for a stored value of ′0′ for 2T and a stored value ′1′ for 3T1D are shown in Figure 69.



(a) 3T1D Retention time of ′1′

(b) 2T Retention time of ′0′

(c) 3T1D Energy-1/Retention product

(d) 2T Energy-1/Retention product

Figure 69: Contour plots for retention time and product of Read MEP energy with 1/retention time. The design with smallest product would be the optimum point with less MEP energy and larger retention time.

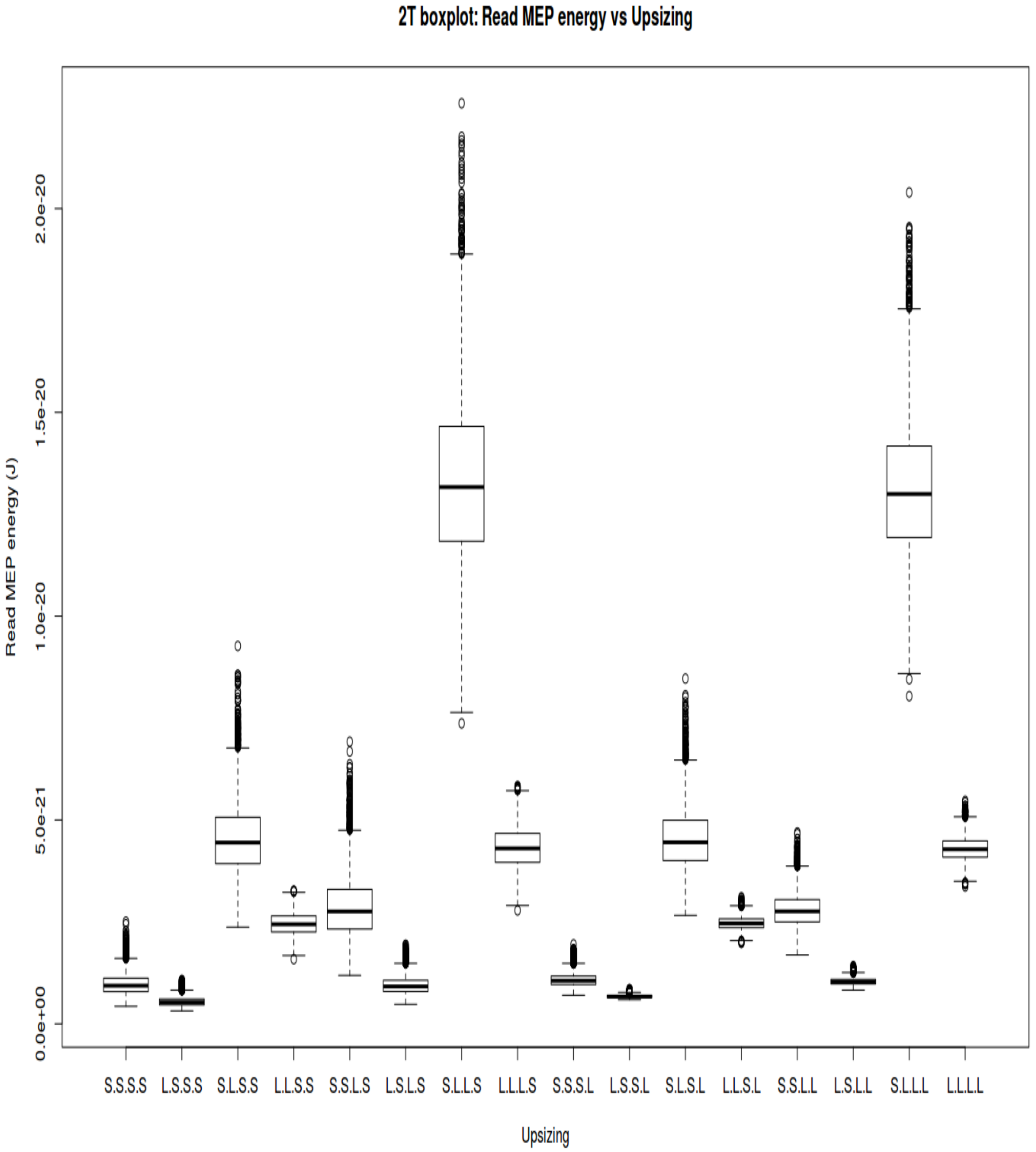
In the case of 3T1D gain cell, the retention time of ′1′ increases with up-sizing of write transistor length up to 2x and then starts decreasing. This is because the MEP supply voltage starts decreasing from 0.18V at 2x length to 0.14V at 4x length write transistor. Though the up-sizing of the read transistor width increases the retention time at a fixed supply voltage, it however decreases the read MEP supply voltage which is 0.18V for 1x width, 0.16V for 2x and 3x width, and 0.14V for 4x read transistor width. The effect of this on retention time is seen in the contour plot in Figure 69a where the retention time of ′1′ at MEP decreases with up-sizing of read transistor width. The HSPICE simulation of the 3T1D design with 2x write transistor length shows 6.6% increase in retention time of ′1′. The contour plot shows that the ′energy * 1/retention time′ product for 3T1D decreases with up-sizing of write transistor length. The HSPICE simulation of design with 4x write transistor length shows 21% decrease in the ′energy * 1/retention time′ product.

In contrast to the retention time of ′1′ in 3T1D, the retention time of ′0′ of 2T increases as MEP supply voltage decreases. The up-sizing of the read transistor width or the write transistor length decreases the MEP supply voltage from 0.18V to 0.1V. The HSPICE simulation of the design with both read transistor width and write transistor length up-sized by 4x shows 25% increase in 2T′s retention time of ′0′. The product ′energy * 1/retention time′ for 2T is higher for the up-sized read transistor width and decreases with up-sizing of write transistor length. The HSPICE simulation of design with 4x up-sized write transistor length shows 44% decrease in this product.

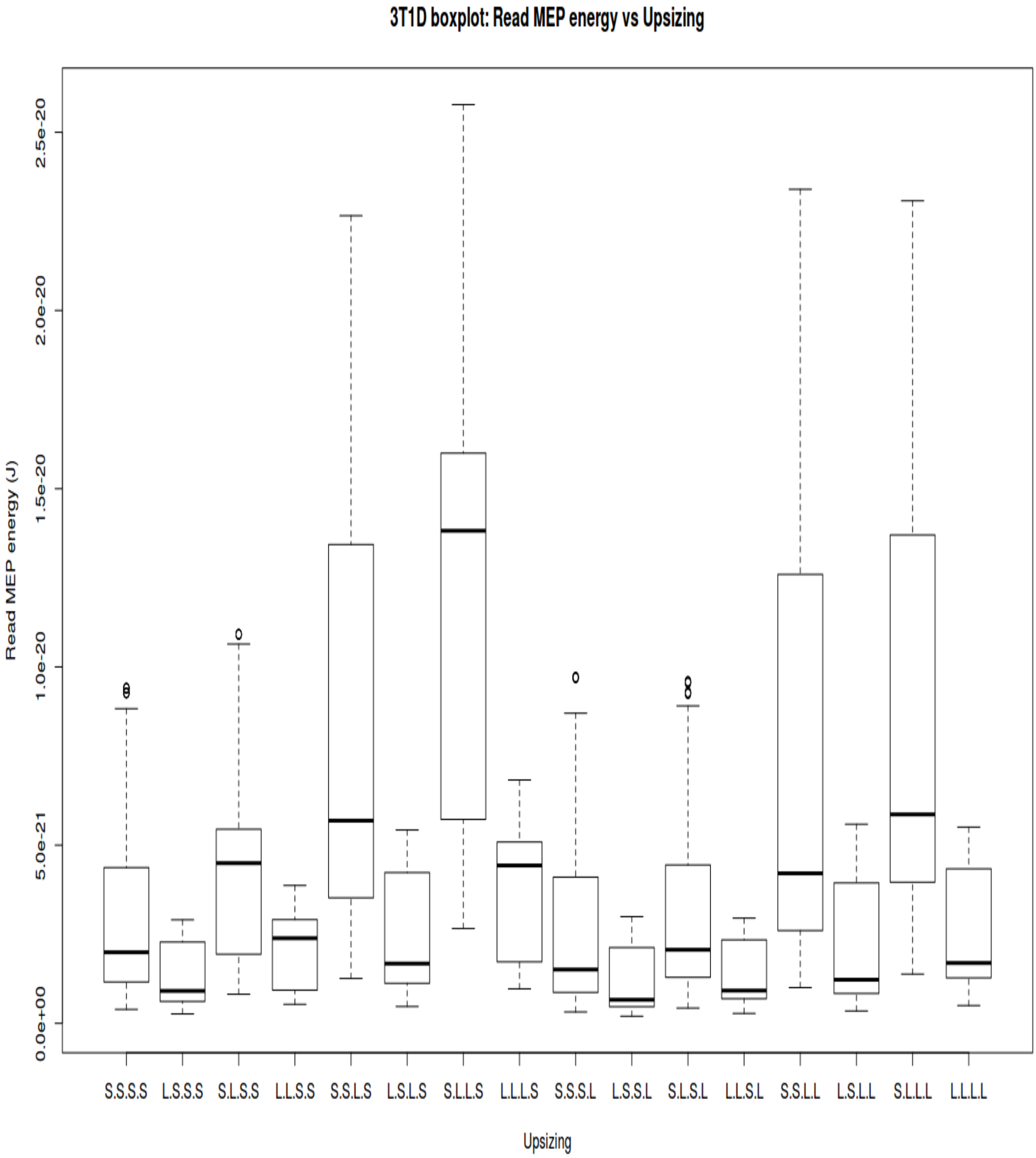
Thus, reducing leakage current through write path by up-sizing the write transistor length also reduces the energy-delay product and the energy-1/retention product. While up-sizing read transistor width to decrease the read delay and increase the retention time, contrarily, increases the energy-delay product and the energy-1/retention product.

6.4.3 Full-Factorial analysis in presence of threshold voltage variations

In presence of process variations, the difference in median MEP energy of different read and write path transistor up-sizing is shown in boxplot Figure 70. For both 2T and 3T1D gain cells, the design with 4x up-sized length for read transistors and width for write transistors (S.L.L.S design) has the maximum median MEP energy. In the case of the 2T gain cell, up-sizing the width of the read transistor has only 12% increase in median MEP energy. The comparison of the 2T gain cell's median MEP energy of the first 8 designs (designs with 1x read transistor width) with the last 8 designs (designs with 4x read transistor width) in the Figure 70 suggests that up-sizing read transistor width does not have significant effect on the median MEP energy.



(a) 2T



(b) 3T1D

Figure 70: Boxplot for MEP energy vs Up-sizing. X-labels are the up-sizing combinations with first two symbols for read and write lengths and last two for widths. "S" is 1x and "L" is 4x increase. Eg, SSSS is the smallest sized design.

To verify this, the p-values from the ANOVA test are calculated for the main effects model. The results are shown in Tables 11 and 12. The p-value in this analysis is interpreted as the probability of observing a difference in the mean MEP energy of 5000 samples for an up-sized design when there is no actual change in MEP energy because of the up-sizing (i.e. the probability of observing different means when the null hypothesis is true) .The effect of an up-sized design on MEP energy is considered to be statistically significant if its p-value is smaller than the significance level, α. Considering the significance level of 0.001 (i.e. less than one in thousand chance of being wrong by rejecting an up-sizing design with significant effect on MEP energy, also called as Type I error). Since the p-value for up-sizing of read transistor width is greater than this significance level, the null hypothesis that up-sizing read transistor width has no statistically significant effect on MEP energy in presence of Vth variations cannot be rejected.

The same conclusion is also reached from the main-effects plot in Figure71 where the 95% confidence intervals of MEP energy for 4x up-sized read transistor width overlap with those of 1x read transistor width.

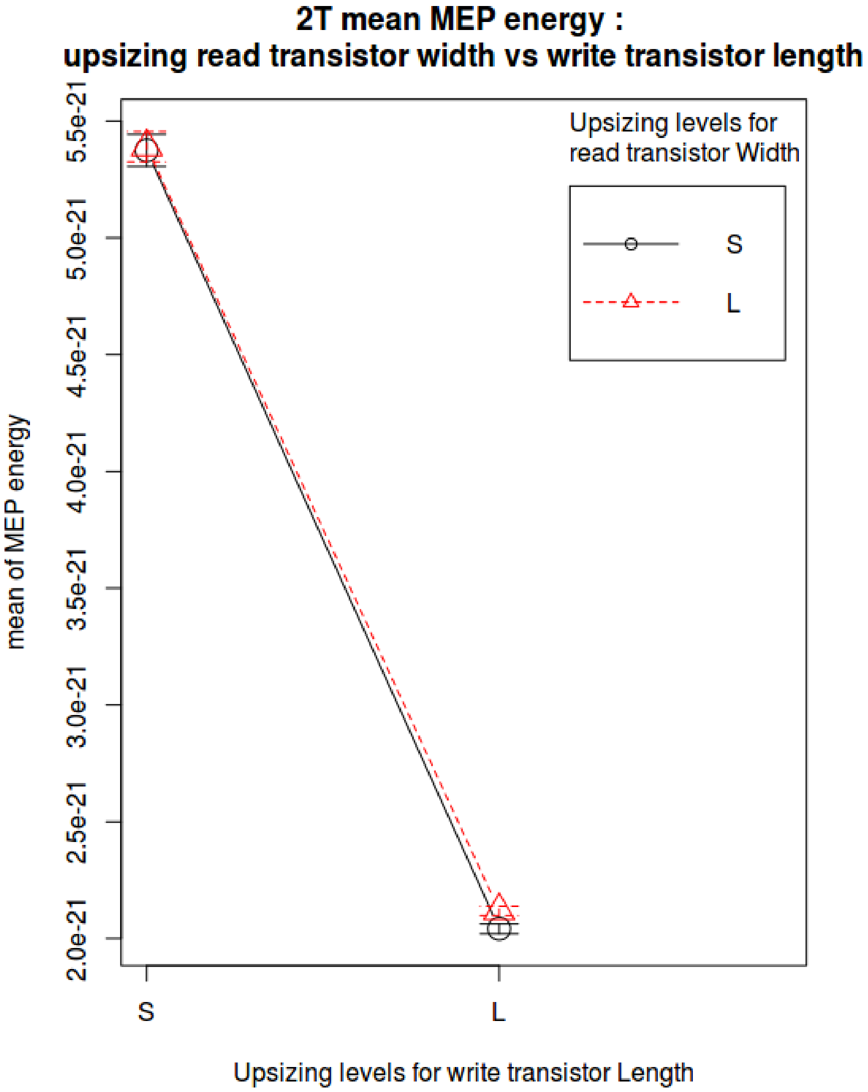


Figure 71: The main effects plot for MEP energy with read transistor width and write transistor length with whiskers as 95%CI. Since the 95% CI for designs with 4x up-sized read transistor width (L) overlap with those of 1x sized read transistor width (S), the null hypothesis that up-sized read transistor width has no statistically significant effect on mean MEP energy in presence of Vth variations, cannot be rejected.

Table 11: p-values for different 2T up-sizing. Smaller p-value means that factor has statistically significant effect. A p-value larger than 0.001 is considered to have no strong statistically significant effect on the response variable.

Up-sizing	p-value
write length	$<2 * 10^{-16}$
write width	$<2 * 10^{-16}$
read length	$<2 * 10^{-16}$
read width	<0.6585

Table 12: p-values for different 3T1D up-sizing.

Up-sizing	p-value
write length	$<2 * 10^{-16}$
read length (T2)	$<2 * 10^{-16}$
read length (T3)	$<2 * 10^{-16}$
diode length	0.8693
write width	$<2 * 10^{-16}$
read width (T2)	$<2 * 10^{-16}$
read width (T3)	$<2 * 10^{-16}$
diode width	0.7525

Table 13: 2T: 95% CI for difference in means of MEP energy between levels :small (1x) and large (4x), for read and write transistors up-sizing. "L" is for large and "S" is for small.

Factor	difference between means of levels	lower 95% CI	upper 95% CI	summary
write length	$\mu(L) - \mu(S)$	$-3.31 * 10^{-21}$	$-3.28 * 10^{-21}$	atleast 60% dec
write width	$\mu(L) - \mu(S)$	$3.14 * 10^{-21}$	$3.17 * 10^{-21}$	atleast 140% inc
read length	$\mu(L) - \mu(S)$	$4.75 * 10^{-21}$	$4.78 * 10^{-21}$	atleast 349% inc

The Tukey's honest significant differences test [Abdi and Williams, 2010] is then used to estimate the set of 95% confidence intervals (CI) of differences between the mean MEP energy between 1x and 4x levels of statistically significant up-sizing factors. The results are shown in Tables 13 and 14. The increase (decrease) in the mean MEP energy at the 4x up-sizing level is calculated as the percentage relative difference between the lower (upper) level value of its 95% CI and the mean at 1x up-sizing level. Up-sizing the write transistor length reduces the mean MEP energy by at-least 60% for 2T and 63% for 3T1D gain cells in presence of threshold voltage variations. The up-sizing factor with largest increase in mean MEP energy in presence of Vth variations for both 2T and 3T1D gain cell is the read transistor length with at least 349% increase for 2T and at least 215% increase for 3T1D.

Table 14: 3T1D: 95% CI for difference in means of MEP energy between levels :small (1x) and large (4x), for read and write transistor up-sizing. "L" is for large and "S" is for small.

Factor	difference between means of levels	lower 95% CI	upper 95% CI	summary
write length	$\mu(L) - \mu(S)$	$-3.87 * 10^{-21}$	$-3.86 * 10^{-21}$	atleast 63% dec
write width	$\mu(L) - \mu(S)$	$3.66 * 10^{-21}$	$3.68 * 10^{-21}$	atleast 160% inc
read length (T2)	$\mu(L) - \mu(S)$	$1.01 * 10^{-21}$	$1.027 * 10^{-21}$	atleast 27% inc
read length (T3)	$\mu(L) - \mu(S)$	$4.30 * 10^{-21}$	$4.32 * 10^{-21}$	atleast 215% inc
read width (T2)	$\mu(L) - \mu(S)$	$-9.02 * 10^{-22}$	$-8.86 * 10^{-22}$	atleast 19% dec
read width (T3)	$\mu(L) - \mu(S)$	$8.62 * 10^{-22}$	$8.78 * 10^{-22}$	atleast 24% inc

6.5 Bootstrap ANOVA

In the previous section, we reported significant up-sizing factors for the MEP read energy using the multi-way ANOVA test. However the distributions of the MEP energy as seen in Figure 70 are not normal and the variances of the different up-sized designs are very different. The assumptions of normality and homoscedasticity are thus not satisfied. Here in Table 15, we provide the results of percentile-t bootstrap one-way ANOVA using trimmed means (20%) for the 2T e-DRAM gain cell to verify that the read transistor width is not a significant factor for read MEP energy. The results are obtained using the “t1waybt" function available in the Wilcox′ robust statistics function R package ([Mair et al., 2015]).

6.6 Conclusion

This chapter discussed the results of our investigation of the minimum read energy operation of 2T and 3T1D gain cell in order to be candidates to substitute SRAM bitcells in sub-threshold memories. Results show that read MEP energy can be reduced by either increasing the length of write transistor (> 26% decrease), or by providing write word-line boosting during read (> 34% decrease), or using high-threshold voltage write transistor (> 25% decrease). In presence of process variations, the p-values from ANOVA show that up-sizing of read transistor width for 2T and up-sizing of diode transistor for 3T1D are not statistically significant factors influencing read MEP energy. The factor resulting in largest increase in read MEP energy for both 2T and 3T1D gain cell is the read transistor length (> 215% increase).

Table 15: Percentile-t one-way Bootstrap ANOVA results vs traditional one-way ANOVA results for 2T gain-cell. Some P-values are zero, that means their numerical values were below the machine precision in R software.

Read Width		
Traditional Anova	F-value=29.779	P-value=$4.9 * 10^{-08}$
Results of Bootstrap Anova (20% trimmed means, Percentile-t)		
# Bootstrap Samples	test-statistic	p-value
100	1.9888	0.19
1000	1.9888	0.159
10000	1.9888	0.1676
* p-value is greater than 0.001, Not statistically significant		

Write Width		
Traditional Anova	F-value=71255	P-value=$2.2 * 10^{-16}$
Results of Bootstrap Anova (20% trimmed means, Percentile-t)		
# Bootstrap Samples	test-statistic	p-value
100	66369.34	0
1000	66369.34	0
10000	66369.34	0
* p-value is less than 0.001, Statistically significant		

Read Length		
Traditional Anova	F-value=97784	P-value=$2.2 * 10^{-16}$
Results of Bootstrap Anova (20% trimmed means, Percentile-t)		
# Bootstrap Samples	test-statistic	p-value
100	91167.24	0
1000	91167.24	0
10000	91167.24	0
* p-value is less than 0.001, Statistically significant		

Write Length		
Traditional Anova	F-value=29059	P-value=$2.2 * 10^{-16}$
Results of Bootstrap Anova (20% trimmed means, Percentile-t)		
# Bootstrap Samples	test-statistic	p-value
100	26793.3	0
1000	26793.3	0
10000	26793.3	0
* p-value is less than 0.001, Statistically significant		

Kids! Check your TV listings. Make sure this isn't the last episode!

Garfield

7 Conclusions and Future Work

7.1 Summary of Contributions

The possibility of achieving maximum energy-efficient memory operation at subthreshold region of operation has motivated the work done in this thesis. However, the fact remains that the memory operation at these ultra-low voltages is marred with vanishing memory noise margins, higher delay and reduced I_{ON}/I_{OFF} ratio. This makes the design of reliable subthreshold memories an arduous undertaking especially with the increased variability in the deep-sub-micron process nodes. Hence, necessitating novel memory bitcell topologies and demanding significant investment of time and computational resources (SPICE simulations) in the yield optimization of these subthreshold memory bitcells.

Furthermore, these subthreshold bitcells have to increase their transistor count compared to six-transistor SRAM bitcell to strengthen their reliability. This either leads to an increase in the dynamic energy per operation or an increase in the number of leakage paths per bitcell, both resulting in increased minimum energy per operation. The rise in the dynamic energy also decreases the MEP voltage which further diminishes the memory noise margins. Likewise, increase in leakage energy pushes MEP to higher supply voltage which can reach above subthreshold voltage range.

The main contributions of the thesis are the following:

1. Reduction in the SPICE simulation cost to find the Most-Probable-Failure-Point (MPFP) which is later used as the shift vector for Mean-Shift Importance Sampling to estimate memory failure probability. In SSFB, we utilized the knowledge that MPFP is at the failure boundary. Hence, random sampling is done only on the surface of hypersphere to keep track of the upper and lower angular coordinates of the failure boundary. This is in contrast to the previous approaches which either randomly sample *within* the volume of an annular region or a hypersphere. Additionally, we proposed REEM method which guides the sampling process to improve the estimates of the failure/ non-failure regions that are likely to sampled under Importance Sampling which results in 10x reduction in the SPICE simulations for the six-transistor SRAM bitcell.

2. Proposal of modeling SRAM memory margins using a new additive kernel made of one-dimensional kernels each encoding the sensitivity information of the memory margin with respect to one of the threshold voltage variation sources and their interaction as product of these one-dimensional kernels. A Gaussian process regression model using the proposed additive kernel provides better extrapolation (32% lower out-of-sample error) than the high dimensional universal kernel function (RBF).

3. Proposal of a novel ten-transistor subthreshold SRAM bitcell with 2x smaller read delay and 54% lower read energy than the previous ten-transistor bitcells. Thus, the bitcell provides an opportunity where the read energy per operation can be further reduced by trading off read delay, for example by using high threshold voltage transistors in the discharge path to reduce leakage current.

4. Characterization of the 2T and 3T1D eDRAM gain cells as alternatives to the SRAM bitcells for their minimum energy operation in size-constrained IoT devices. We show that the energy efficiency at the read MEP of the eDRAM gain cells is increased by reducing the leakage current through write path (using high threshold voltage transistor, write wordline boosting or upsizing transistor length). While reducing the read delay (using read wordline boosting, upsizing transistor width) to decrease the leakage energy per operation, on the contrary, increases read MEP energy.

7.2 Future Work

1. The Importance Sampling based techniques have been the central focus for the memory failure probability estimation part of this thesis. The two proposals SSFB and REEM, both are motivated with the need to find the Most-Probable-Failure-Point (MPFP) faster, which is to be used as the shift vector in mean-shift Importance Sampling. However, as it was highlighted in the background chapter, the accuracy of Importance Sampling does not scale with increase in the number of variation sources. There exists enhanced Importance Sampling techniques for rare event simulations such as Annealed Importance Sampling (AIS) which assigns weights based on simulated annealing scheme. There are also recent developments in the use of population based Markov-Chain Monte Carlo (PMCMC) methods such as "Parallel Tempering" to explore high dimensional multi-model parameter spaces. In these methods, an ensemble of distributions is used to model the target distribution and a population of samples is made by sampling from each distribution of the ensemble. The difference from traditional MCMC methods such as Metropolis-Hasting sampling is that instead of a single MCMC chain there are several MCMC chains simulating in parallel and information is allowed to propagate through these chains by swapping states among chains based on the Metropolis criterion. These techniques can be investigated for application in the memory yield estimation under larger number of variation sources.

2. The surrogate modeling of the memory margins in this thesis was done using Gaussian process regression. The objective there are was to predict the memory margins accurately away from the region sampled using Latin Hypercube Sampling (LHS). However, we noticed there that the models could not predict the read/write margin failure regions accurately and we had to sample near the MPFP to add more failure samples to the training set. A better approach would be to sequentially create the training set by targeting only the failure region. The recent developments in the Bayesian optimization of expensive functions provide this opportunity. The general problem in this field is of exploration-vs-exploitation trade-off. Since the function (in our case the memory failure indicator function) is unknown, the task can be described as finding the optimal fraction of samples for the exploration of failure regions in the parameter space and the fraction of samples to be sampled from these current estimates of failure regions to increase the model accuracy in these failure regions. The Bayesian bandit algorithms such as the Expected Improvement method and the GP-UCB method are currently the state-of-the-art and can be investigated for building the surrogate models of failure regions of memory margins.

3. While analysis of the SRAM bitcells and eDRAM gain cells in this thesis was based on conventional CMOS technology, new process technologies such as FinFET, PDSoI/FDSoI should also be studied because the shift to one of these technologies is inevitable in the near future

7.3 Publications

1. *"SSFB: A highly-efficient and scalable simulation reduction technique for SRAM yield analysis."* Manish Rana, and Ramon Canal. IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014.

2. *"REEM: Failure/non-failure region estimation method for SRAM yield analysis."* Manish Rana, and Ramon Canal. 32nd IEEE International Conference on Computer Design (ICCD), 2014.

3. *"Statistical Analysis and Comparison of 2T and 3T1D e-DRAM Minimum Energy Operation."* Manish Rana, Ramon Canal, Esteve Amat and Antonio Rubio. IEEE 22st International On-Line Testing Symposium (IOLTS), 2016.

4. (Under submission) *"SRAM Memory Margin Probability Failure Estimation using Gaussian Process Regression."* Manish Rana, Ramon canal, Jie Han, Bruce Cockburn. Submitted to the 34th IEEE International Conference on Computer Design (ICCD), 2016

5. (Journal, under submission) *"Minimum Energy Analysis of the e-DRAM gain cells to achieve robust minimum energy operation"*. Manish Rana, Ramon Canal, Esteve Amat, and Antonio Rubio. IEEE Transactions on Device and Materials Reliability (TDMR).

6. (Journal, under preparation) *"Worst PVT corner prediction for memory circuits using conditioned Gaussian process regression"*

Bibliography

[HSP,] https://www.synopsys.com.

[PTM,] http://ptm.asu.edu/.

[Abdi and Williams, 2010] Abdi, H. and Williams, L. J. (2010). Tukey's honestly significant difference (hsd) test. *Encyclopedia of Research Design. Thousand Oaks, CA: Sage*, pages 1–5.

[Abu-Mostafa et al., 2012] Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. (2012). *Learning from data*. AMLBook Berlin, Germany.

[Amat et al., 2014] Amat, E., Calomarde, A., Moll, F., Canal, R., and Rubio, A. (2014). Feasibility of the embedded dram cells implementation withfinfet devices. *Computers, IEEE Transactions on*.

[Amat et al., 2015] Amat, E., Christmann, J.-F., Billoint, O., Miro, I., and Beigne, E. (2015). Fdsoi suitability for asynchronous circuits at sub-vt. *ECS Transactions*, 66(5):315–322.

[BAI et al., 2011] BAI, N., WU, X., YANG, J., and SHI, L. (2011). A robust high density 7t sram bitcell for subthreshold applications. *Chinese Journal of Electronics*, 20(2):243–246.

[Banerjee and Calhoun, 2014] Banerjee, A. and Calhoun, B. H. (2014). An Ultra-Low Energy Subthreshold SRAM Bitcell for Energy Constrained Biomedical Applications. *Journal of Low Power Electronics and Applications*, 4(2):119–137.

[Bansal et al., 2009] Bansal, A., Rao, R., Kim, J.-J., Zafar, S., Stathis, J. H., and Chuang, C.-T. (2009). Impact of nbti and pbti in sram bit-cells: Relative sensitivities and guidelines for application-specific target stability/performance. In *Reliability Physics Symposium, 2009 IEEE International*, pages 745–749. IEEE.

[Bol et al., 2013] Bol, D., De Vos, J., Hocquet, C., Botman, F., Durvaux, F., Boyd, S., Flandre, D., and Legat, J. (2013). Sleepwalker: A 25-mhz 0.4-v sub-7-microcontroller in 65-nm lp/gp cmos for low-carbon wireless sensor nodes. *Solid-State Circuits, IEEE Journal of*, 48(1):20–32.

[Boley et al., 2012] Boley, J., Wang, J., and Calhoun, B. H. (2012). Analyzing sub-threshold bitcell topologies and the effects of assist methods on sram vmin. *Journal of Low Power Electronics and Applications*, 2(2):143–154.

[Box et al., 1978] Box, G. E., Hunter, W. G., Hunter, J. S., et al. (1978). Statistics for experimenters.

[Calhoun and Chandrakasan, 2004] Calhoun, B. H. and Chandrakasan, A. (2004). Characterizing and modeling minimum energy operation for subthreshold circuits. In *Low Power Electronics and Design, 2004. ISLPED'04. Proceedings of the 2004 International Symposium on*, pages 90–95. IEEE.

[Calhoun and Chandrakasan, 2006a] Calhoun, B. H. and Chandrakasan, A. P. (2006a). Static noise margin variation for sub-threshold SRAM in 65-nm CMOS. *Solid-State Circuits, IEEE Journal of*, 41(7):1673–1679.

[Calhoun and Chandrakasan, 2006b] Calhoun, B. H. and Chandrakasan, A. P. (2006b). Static noise margin variation for sub-threshold sram in 65-nm cmos. *Solid-State Circuits, IEEE Journal of*, 41(7):1673–1679.

[Calhoun and Chandrakasan, 2007] Calhoun, B. H. and Chandrakasan, A. P. (2007). A 256-kb 65-nm sub-threshold sram design for ultra-low-voltage operation. *Solid-State Circuits, IEEE Journal of*, 42(3):680–688.

[Calhoun et al., 2005] Calhoun, B. H., Wang, A., and Chandrakasan, A. (2005). Modeling and sizing for minimum energy operation in subthreshold circuits. *Solid-State Circuits, IEEE Journal of*, 40(9):1778–1786.

[Chang et al., 2009] Chang, I. J., Kim, J.-J., Park, S. P., and Roy, K. (2009). A 32 kb 10t subthreshold sram array with bit-interleaving and differential read scheme in 90 nm cmos. *Solid-State Circuits, IEEE Journal of*, 44(2):650–658.

[Chang et al., 2011] Chang, M.-F., Chang, S.-W., Chou, P.-W., and Wu, W.-C. (2011). A 130 mv sram with expanded write and read margins for subthreshold applications. *Solid-State Circuits, IEEE Journal of*, 46(2):520–529.

[Chang et al., 2010] Chang, M.-F., Wu, J.-J., Chen, K.-T., Chen, Y.-C., Chen, Y.-H., Lee, R., Liao, H.-J., and Yamauchi, H. (2010). A differential data-aware power-supplied (d ap) 8t sram cell with expanded write/read stabilities for lower vddmin applications. *Solid-State Circuits, IEEE Journal of*, 45(6):1234–1245.

[Chang et al., 2012] Chang, M.-H., Chiu, Y.-T., and Hwang, W. (2012). Design and iso-area analysis of 9t subthreshold sram with bit-interleaving scheme in 65-nm cmos. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 59(7):429–433.

[Chiu et al., 2014] Chiu, Y.-W., Hu, Y.-H., Tu, M.-H., Zhao, J.-K., Chu, Y.-H., Jou, S.-J., and Chuang, C.-T. (2014). 40 nm bit-interleaving 12t subthreshold sram with data-aware write-assist. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 61(9):2578–2585.

[Chun et al., 2009] Chun, K. C., Jain, P., Lee, J. H., and Kim, C. H. (2009). A sub-0.9 v logic-compatible embedded dram with boosted 3t gain cell, regulated bit-line write scheme and pvt-tracking read reference bias. In *VLSI Circuits, Symposium on*.

[Chun et al., 2011] Chun, K. C., Jain, P., Lee, J. H., and Kim, C. H. (2011). A 3t gain cell embedded dram utilizing preferential boosting for high density and low power on-die caches. *Solid-State Circuits, IEEE Journal of*.

[Dogan et al., 2012] Dogan, A. Y., Constantin, J., Ruggiero, M., Burg, A., and Atienza, D. (2012). Multi-core architecture design for ultra-low-power wearable health monitoring systems. In *DATE*, pages 988–993.

[Dolecek et al., 2008] Dolecek, L., Qazi, M., Shah, D., and Chandrakasan, A. (2008). Breaking the simulation barrier: Sram evaluation through norm minimization. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 322–329. IEEE Press.

[Doorn et al., 2008] Doorn, T., Maten, E., Croon, J., Bucchianico, A. D., and Wittich, O. (2008). Importance sampling monte carlo simulations for accurate estimation of sram yield. In *Solid-State Circuits Conference, 2008. ESSCIRC 2008. 34th European*, pages 230–233. IEEE.

[Dupuy et al., 2015] Dupuy, D., Helbert, C., and Franco, J. (2015). DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):1–38.

[Duvenaud, 2014] Duvenaud, D. (2014). *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge.

[Duvenaud et al., 2011] Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. (2011). Additive gaussian processes. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 226–234. Curran Associates, Inc.

[Friedli, 2016] Friedli, Kaufmann, P. K. (2016). Energy Efficiency of the Internet of Things : Technology and Energy Assessment Report. *IEA 4E EDNA*.

[Geenens et al., 2011] Geenens, G. et al. (2011). Curse of dimensionality and related issues in nonparametric functional regression. *Statistics Surveys*, 5:30–43.

[Hagiwara et al., 2010] Hagiwara, S., Masu, K., Sato, T., et al. (2010). Robust importance sampling for efficient sram yield analysis. In *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, pages 15–21. IEEE.

[Hanson et al., 2006] Hanson, S., Zhai, B., Bernstein, K., Blaauw, D., Bryant, A., Chang, L., Das, K. K., Haensch, W., Nowak, E. J., and Sylvester, D. M. (2006). Ultralow-voltage, minimum-energy CMOS. *IBM journal of research and development*, 50(4.5):469–490.

[Hesterberg, 2003] Hesterberg, T. C. (2003). *Advances in importance sampling*. PhD thesis, Stanford University.

[Iqbal et al., 2012] Iqbal, R., Meinerzhagen, P., and Burg, A. (2012). Two-port low-power gain-cell storage array: voltage scaling and retention time. In *ISCAS*. IEEE.

[Kanj et al., 2006] Kanj, R., Joshi, R., and Nassif, S. (2006). Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events. In *Proceedings of the 43rd annual Design Automation Conference*, pages 69–72. ACM.

[Kanj et al., 2007] Kanj, R., Joshi, R., Sivagnaname, J., Kuang, J. B., Acharyya, D., Nguyen, T., McDowell, C., and Nassif, S. (2007). Gate leakage effects on yield and design considerations of pd/soi sram designs. In *Quality Electronic Design, 2007. ISQED'07. 8th International Symposium on*, pages 33–40. IEEE.

[Katayama et al., 2010] Katayama, K., Hagiwara, S., Tsutsui, H., Ochi, H., and Sato, T. (2010). Sequential importance sampling for low-probability and high-dimensional sram yield analysis. In *Proceedings of the International Conference on Computer-Aided Design*, pages 703–708. IEEE Press.

[Khalil et al., 2008] Khalil, D., Khellah, M., Kim, N.-S., Ismail, Y., Karnik, T., and De, V. K. (2008). Accurate estimation of sram dynamic stability. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(12):1639–1647.

[Kida et al., 2012] Kida, T., Tsukamoto, Y., and Kihara, Y. (2012). Optimization of importance sampling monte carlo using consecutive mean-shift method and its application to sram dynamic stability analysis. In *Quality Electronic Design (ISQED), 2012 13th International Symposium on*, pages 572–579. IEEE.

[Kim et al., 2011] Kim, D., Chen, G., Fojtik, M., Seok, M., Blaauw, D., and Sylvester, D. (2011). A 1.85 fw/bit ultra low leakage 10t sram with speed compensation scheme. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 69–72. IEEE.

[Kim et al., 2007a] Kim, T.-H., Liu, J., Keane, J., and Kim, C. (2007a). A high-density subthreshold sram with data-independent bitline leakage and virtual ground replica scheme. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pages 330–606.

[Kim et al., 2007b] Kim, T.-H., Liu, J., and Kim, C. (2007b). An 8t subthreshold sram cell utilizing reverse short channel effect for write margin and read performance improvement. In *Custom Integrated Circuits Conference, 2007. CICC '07. IEEE*, pages 241–244.

[Kulkarni et al., 2007] Kulkarni, J. P., Kim, K., and Roy, K. (2007). A 160 mv robust schmitt trigger based subthreshold sram. *Solid-State Circuits, IEEE Journal of*.

[Kulkarni and Roy, 2012] Kulkarni, J. P. and Roy, K. (2012). Ultralow-voltage process-variation-tolerant schmitt-trigger-based sram design. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 20(2):319–332.

[Liang et al., 2008] Liang, X., Canal, R., Wei, G.-Y., and Brooks, D. (2008). Replacing 6t srams with 3t1d drams in the l1 data cache to combat process variability. *IEEE micro*.

[Lo and Huang, 2011] Lo, C.-H. and Huang, S.-Y. (2011). Ppn based 10t sram cell for low-leakage and resilient subthreshold operation. *Solid-State Circuits, IEEE Journal of*, 46(3):695–704.

[Lovin et al., 2009] Lovin, K., Lee, B. C., Liang, X., Brooks, D., and Wei, G.-Y. (2009). Empirical performance models for 3t1d memories. In *ICCD*, pages 398–403. IEEE.

[Mair et al., 2015] Mair, P., Schoenbrodt, F., and Wilcox, R. (2015). *WRS2: Wilcox robust estimation and testing*. 0.4-0.

[Manyika et al., 2011] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity.

[McConaghy, 2011] McConaghy, T. (2011). Ffx: Fast, scalable, deterministic symbolic regression technology. In *Genetic Programming Theory and Practice IX*, pages 235–260. Springer.

[Meinerzhagen et al., 2013] Meinerzhagen, P., Teman, A., Giterman, R., Burg, A., and Fish, A. (2013). Exploration of sub-vt and near-vt 2t gain-cell memories for ultra-low power applications under technology scaling. *Journal of Low Power Electronics and Applications*.

[Micchelli et al., 2006] Micchelli, C. A., Xu, Y., and Zhang, H. (2006). Universal kernels. *The Journal of Machine Learning Research*, 7:2651–2667.

[NXP,] NXP. Lpc1102/1104 product data sheet. Available at: http://www.nxp.com/documents/data sheet/LPC1102 1104.pdf.

[Okobiah et al., 2014] Okobiah, O., Mohanty, S., and Kougianos, E. (2014). Fast design optimization through simple kriging metamodeling: A sense amplifier case study. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(4):932–937.

[Pronzato and Müller, 2012] Pronzato, L. and Müller, W. G. (2012). Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701.

[Qazi et al., 2010] Qazi, M., Tikekar, M., Dolecek, L., Shah, D., and Chandrakasan, A. (2010). Loop flattening & spherical sampling: highly efficient model reduction techniques for sram yield analysis. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 801–806. European Design and Automation Association.

[Rasmussen, 2006] Rasmussen, C. E. (2006). Gaussian processes for machine learning.

[Raychowdhury et al., 2005] Raychowdhury, A., Mukhopadhyay, S., and Roy, K. (2005). A feasibility study of subthreshold SRAM across technology generations. In *Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference on*, pages 417–422. IEEE.

[Rubio Sola et al., 2012] Rubio Sola, J. A., Figueras Pàmies, J., Vatajelu, E. I., and Canal Corretger, R. (2012). Process variability in sub-16nm bulk cmos technology.

[Singh et al., 2008] Singh, J., Mathew, J., Pradhan, D. K., and Mohanty, S. P. (2008). A subthreshold single ended i/o sram cell design for nanometer cmos technologies. In *SOC Conference, 2008 IEEE International*, pages 243–246. IEEE.

[Singhee and Rutenbar, 2008] Singhee, A. and Rutenbar, R. A. (2008). Statistical blockade: a novel method for very fast monte carlo simulation of rare circuit events, and its application. In *Design, Automation, and Test in Europe*, pages 235–251. Springer.

[Swanson and Meindl, 1972] Swanson, R. M. and Meindl, J. D. (1972). Ion-implanted complementary mos transistors in low-voltage circuits. *Solid-State Circuits, IEEE Journal of*, 7(2):146–153.

[Teman et al., 2012] Teman, A., Meinerzhagen, P., Burg, A., and Fish, A. (2012). Review and classification of gain cell edram implementations. In *Electrical & Electronics Engineers in Israel (IEEEI), IEEE 27th Convention of*.

[Verma and Chandrakasan, 2008] Verma, N. and Chandrakasan, A. P. (2008). A 256 kb 65 nm 8t subthreshold sram employing sense-amplifier redundancy. *Solid-State Circuits, IEEE Journal of*, 43(1):141–149.

[Walker, 2013] Walker, S. (2013). Wearable Technology-Market Assessment: An IHS Whitepaper. *IHS Electronics*.

[Wang et al., 2008] Wang, J., Nalam, S., and Calhoun, B. H. (2008). Analyzing static and dynamic write margin for nanometer srams. In *Low Power Electronics and Design (ISLPED), 2008 ACM/IEEE International Symposium on*, pages 129–134. IEEE.

[Wicht et al., 2004] Wicht, B., Nirschl, T., and Schmitt-Landsiedel, D. (2004). Yield and speed optimization of a latch-type voltage sense amplifier. *Solid-State Circuits, IEEE Journal of*, 39(7):1148–1158.

[Wilson, 2014] Wilson, A. G. (2014). *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, PhD thesis, University of Cambridge.

[Wu et al., 2014] Wu, W., Xu, W., Krishnan, R., Chen, Y.-L., and He, L. (2014). Rescope: High-dimensional statistical circuit simulation towards full failure region coverage. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6.

[Yao et al., 2013] Yao, J., Ye, Z., and Wang, Y. (2013). Efficient importance sampling for high-sigma yield analysis with adaptive online surrogate modeling. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pages 1291–1296.

[Zhao et al., 2015] Zhao, Y., Shin, H., Chen, H., Tan, S. X. D., Shi, G., and Li, X. (2015). Statistical rare event analysis using smart sampling and parameter guidance. In *2015 28th IEEE International System-on-Chip Conference (SOCC)*, pages 53–58.