Abstract

This project has two main aspects. The first one is using the method of Statistical Blockade to analyze of PVT variation in SRAM cells. As the feature size of the device circuits decreases, the PVT variation becomes an increasing concern. Furthermore, this variability will impact the performance of the integrated circuit. On the other hand, as the SRAM cell is a high-replication circuit, and there would be millions of memory cells in one chip. Therefore, a rare event of the SRAM cell may cause a not-so-rare failure of the whole system. Although the standard Monte Carlo method is an accurate method of simulating, it is too slow to sample the rare and critical events of such high-replication circuit. Due to this situation, an efficient approach of simulating the rare circuit event on SRAM is required in terms of providing high speed simulating. It could be achieved by using the method of Statistical Blockade.

The second part of the project is the low leakage design of SRAM cell with dual threshold voltage transistors. Recently, the Static Power of the circuit plays an important role in the total power consumption. Due to technology scaling, to reduce the Static Power is important in the design of VLSI systems. In this project, the approach of dual threshold voltage is carried out to reduce the leakage power of the SRAM cell. There are two kinds of threshold voltage transistors in the SRAM cell, one with the normal value of threshold voltage, the other one with the high threshold voltage. Within this project, there are three configurations of transistors has been proposed with the measurements of the leakage power respectively. Meanwhile, within each simulation, the read delay, write delay, and the Static Noise Margin are estimated as well. The results show that the approach of the dual threshold voltage does not only reduce the leakage power of the SRAM cell, but also improve the Static Noise Margin.

The main contributions and achievements of this project are shown as follows:

- The algorithm of the Statistical Blockade method is implemented in this project, and it is adapted as a novel approach of simulating rare circuit events.
- The idea of further speeding up the simulation of rare circuit events is also produced by minimizing the classification threshold and training samples in Statistical Blockade approach.
- Research is carried out to discover the changes in the minimal classification threshold and the minimal training samples with different deviations of parameters.
- The low leakage design of an SRAM cell with dual threshold voltage is accomplished in this project.

Acknowledgements

I would like to thank my supervisors Professor Prof. Dhiraj K. Pradhan, Dr. Jimson Mathew, for their continuous support with this research, their comments, assistance and encouragement has proved throughout this project.

Thanks also to Luo Sun for his help, support, and feedback for this research.

Thanks to Yifan Zhang for the support with the software used in this project.

I would like to thank Zhijun Zhou for reading this report.

Na Zhang, September 2011

Content

Abstract I

Acknowledgements II

1. Introduction 1

2. Aims and objectives 2

2.1 Aims and objectives 2

2.2 The structure of this report 2

3. Background 3

3.1 PVT variation 3

3.1.1 Sources of variability 3

3.1.2 Solution of the variation trend 5

3.2 SRAM cell 5

3.2.1 Classical 6T SRAM 6

3.2.2 Working principle of 6T SRAM 7

3.2.3 Three operation states of 6T SRAM 7

3.3 Statistical analysis 8

3.3.1 Corner analysis 8

3.3.2 Statistical model vs. corner model 9

3.3.3 NMOS variation and PMOS variation determination 9

3.3.4 Key elements of statistical circuit analysis 10

3.3.4.1 Extraction of statistical device models 10

3.3.4.2 Sensitivity analysis 11

3.3.4.3 Worst-case analysis 11

3.3.4.4 Spatial modeling and mismatch analyses 11

3.4 Circuit level simulation using Monte Carlo 12

3.5 Statistical Blockade 13

3.5.1 The basic idea of calling Statistical Blockade 14

3.5.1.1 Use the Extreme Value Theory to model the rare events 15

3.5.1.2 Fitting the model and prediction 15

3.5.2 The implementation of the Statistical Blockade 15

3.6 Low-leakage SRAM design with dual Vt transistors ... 16
3.6.1 The power consumption of a circuit ... 16
3.6.1 The idea of dual threshold voltage design ... 17
3.7 Static Noise Margin ... 18
3.7.2 The concept of static noise margin ... 18
3.7.3 The calculation of Static Noise Margin ... 18
4. Methods ... 21
4.1 Analysis using the Statistical Blockade method ... 21
4.1.1 The algorithm of Statistical Blockade method ... 21
4.1.2 The algorithm of finding the minimal classification threshold ... 22
4.1.3 The algorithm of finding the minimal training samples ... 24
4.2 Different deviations of parameters ... 25
4.3 Low leakage SRAM design ... 26
4.4 Static Noise Margin ... 28
5. Results ... 31
5.1 Analysis of PVT variation using the Standard Monte Carlo ... 31
5.2 Results for Statistical Blockade ... 36
5.2.1 The idea of Statistical Blockade ... 36
5.2.2 Comparing the Statistical Blockade with the Standard Monte Carlo Simulation.... 38
5.3 Evaluate the minimal classification threshold and training samples ... 39
5.4 Different deviations of parameters with Statistical Blockade ... 40
5.5 Low leakage SRAM design with dual threshold voltage transistors ... 41
6. Critical Evaluation ... 46
7. Future work ... 48
7.1 Statistical Blockade method ... 48
7.2 Low-leakage design of SRAM cell with Dual **Vt** transistors ... 48
8. Conclusion ... 49
9. Bibliography ... 50
Appendix: Source code ... 53

Figures and Tables

Fig 3. 1 Device and interconnect variation trends for different technology generations. 4
Fig 3. 2 frequency probability and power consumption in different transistor size 5
Fig 3. 3 The schematic of a 6-T SRAM cell 6
Fig 3. 4 The read operation of a 6-T SRAM cell (read 0) 8
Fig 3. 5 The procedure of Monte Carlo simulation 13
Fig 3. 6 Millions of memory cells on the chip [16]...14
Fig 3. 7 The idea of Statistical Blockade (Figure from [3]) 16
Fig 3. 8 A flip-flop comprised of two inverters. Static-noise voltage sources V, are included........18
Fig 3. 9 Graphical representation of the Static Noise Margin (Figure from [26]) 19

Fig 4. 1 The algorithm of the Statistical Blockade method [3] 21
Fig 4. 2 The algorithm of choosing the minimal classification threshold [31] 23
Fig 4. 3 The algorithm for choosing the minimal training sample[31] 24
Fig 4. 4 The Gaussian distribution with higher deviation [32]..25
Fig 4. 5 The schematic of SRAM cell in hold state 26
Fig 4. 6 The schematic of SRAM cell in read operation..27
Fig 4.7 The schematic of SNM simulation in standby state..29
Fig 4.8 The SNM in standby state...29
Fig 4. 9 The schematic of SNM simulation in read state 29
Fig 4. 10 The SNM in read state 29
Fig 4. 11 The SNM of write state...30
Fig 4. 12 The schematic of SNM simulation in write state 30

Fig 5. 1 The distribution of the threshold voltage of NMOS 32
Fig 5. 2 The results of read time and write time with one parameter variation...............................32
Fig 5. 3 The distributions for threshold voltage of NMOS and PMOS..33
Fig 5. 4 The simulation results of read time and write time of the SRAM with two parameter variations...33
Fig 5. 5 The distributions of threshold voltage of NMOS, PMOS, and the gate oxide thickness of NMOS 34
Fig 5. 6 The simulation results for the read time and write time of the SRAM cell under three parameter variations 34
Fig 5. 7 The distributions of threshold voltage of NMOS, PMOS and the gate oxide thickness of NMOS, PMOS 35
Fig 5. 8 The results of read time and write time of the SRAM cell under four parameter variations 36
Fig 5. 9 Training the classifier 37
Fig 5. 10 The Classification of Generated Samples 37
Fig 5. 11 Empirical vs. Statistical Blockade 38
Fig 5. 12 The reduction of leakage current for each configuration in standby state 42
Fig 5. 13 The reduction of leakage current for each configuration in read operation 42

Fig 5. 14 The reduction of leakage current for each configuration in read operation 43
Fig 5. 15 The read time for each configuration in standby operation 44
Fig 5. 16 The write time for each configuration in standby operation 44
Fig 5. 17 The static noise margin for each configuration in standby operation 45
Fig 5. 18 The SNM of each configuration in standby state..45

Table 1 The different configurations of low leakage design 28
Table 2 The simulations times of Standard MC and SB and the tail points they covered..............39
Table 3 The minimal classification threshold and training samples for different tail threshold......40
Table 4 True threshold of the tail points, minimal classification threshold, and minimal training sample required for different deviations of parameter 41

1. Introduction

PVT variation plays an important role in the performance of integrated circuits. PVT refers to process(P) parameters, power supply voltages(V) and operating temperatures(T). The variations refer to the deviations from designed parameter values. As the current or future feature size of the integrated circuit is being reduced, the parameter variations are increasing. Moreover, the circuits like SRAM or flip-flop are high-replication, there may be millions of the cells in one chip. Therefore, the rare event of the cell would result in a not-so-rare failure probability of the system. It is necessary to find a way to simulate the rare circuit events efficiently. As we want to meet the further development of this field, it is necessary to analyze the PVT variations in SRAM cells and develop a new way to detect the rare events of the circuit.

Standard Monte Carlo is an accurate way to simulate the circuit. However, the drawback of the simulation is cost of time, and there are large amount of the simulations required to sample the rare events of the circuit like SRAM, which will cost longer time. Compared to conventional methodology, the Statistical Blockade approach is able to perform a more efficient way to sample and analyze the rare circuit events.

Associated with its high performance and wide bus-width capability, embedded memory becomes an essential component in VLSI. Embedded memory is one kind of integrated on-chip memory which can support the core on chip to accomplish particular functions and communication between cores [1]. It is a particular circuit which is high replicated. In this project, the analysis of the PVT variation in memory cell using Statistical Blockade method, particularly on SRAM cell. By implementing this method, an efficient simulation of the rare circuit events on SRAM cell could be achieved. We will also propose a way to further speed up the simulation of rare events based on the idea of Statistical Blockade.

For integrated circuits design, the power consumption is one of the most essential issues. It can be divided into two categories, which refers to dynamic power and static power. At the initial stages of chip design, the dynamic power is the dominant part of the total power dissipation. Refer to technique scaling, the Static Power is increasingly important. In this project, a low leakage design of the SRAM cell with the dual threshold voltage is tested, in order to reduce the static power [2].

The general experiments provided in this project are aimed to analyse the PVT variation by using Statistical Blockade technique and manipulate a low leakage design with dual threshold voltage.

2. Aims and objectives

This section will express the main aims and the objectives of this project. Further, the structure of this dissertation will be explained.

2.1 Aims and objectives

The aim of this project is to compare the efficiency between conventional and Statistical Blockade approach in order to provide a faster rare circuit event under process variation. Based on the Statistical Blockade approach which proposed by Singhee [3], the idea of further speed up the simulation of rare circuit event will be carried out. Also, it aims to propose a low leakage power design and investigate the effects of Static Noise Margin for the SRAM cell.

The objectives of this project are going to address following questions:

- What is the problem with the Standard Monte Carlo technique for simulating the rare circuit events? Analysis of PVT variation on SRAM cell using Statistical Blockade method, and compare this approach with standard MC.
- Can we further speed up the simulation of rare circuit events based on the method of Statistical Blockade? Analyze the idea of minimizing the classification threshold and training samples of the classifier.
- If we widen the deviation of the Gaussian distribution for each parameter, what will happen on the value of minimal classification threshold and training samples.
- How to reduce the leakage power of the SRAM cell? Propose a low leakage design on SRAM cell with the dual threshold voltage.

2.2 The structure of this report

There are four main sections in this report. Firstly, the background of the field which this project is proposed is explained in chapter three. The sources of the PVT variation are analyzed, and the consideration of variation tolerance design is proposed. The working principle and the operations of the SRAM which is the test circuit of our project are expressed in this section. We also explain the idea of Statistical Blockade method. The low leakage design is also proposed in this section. Secondly, the methods which are used in this project and the works carried out are presented in chapter four. It includes the detail description of how the method work, the problems we met and how we solve them. Next, the result section presents the results of the project we got and the discussion of each objective question. Then a critical evaluation of the work we have done is explained, and we will discuss whether the work we have done met the initial objectives. Finally, the future work related to the area of this project will be discussed in the last section.

3. Background

3.1 PVT variation

PVT variation refers to process, voltage, and temperature variation. Process variation results from the limitations of manufacturing in nanoscale technologies, variation in voltage due to the various power supplies, and temperature variation is because of the manufacturing environment [4]. These variations present a negative impact on the device quality.

In some research, the process, voltage, and temperature variation can be investigated and controlled by their individual systems independently. For instance, the minimum consumption of Vdd can be created for various temperature value and different libraries by checking them in lookup table depend on the PVT evaluated and controlled system [5].

As the PVT variations are increasingly concerned, it becomes a crucial issue for circuit design. Moreover, the integrated circuit design is scaling down, hence, how to control the variation of each parameter is an important part of circuit design. In this situation, generating a series of design techniques to minimize the effects of these parameter variations on the device performance is necessary. In this project, we will propose the particular circuit level model which refers to SRAM to analyze the impact of PVT variations on nanometer technologies, that is specifically 32 nm and below. Then compare this simulation result of circuit level model with the existing behavioral model.

3.1.1 Sources of variability

There are two sources of variation which can influence the electrical performance of the integrated circuits. One kind of sources are environmental factors, it is caused by the specifics of the design implementation. These factors include changes in power supply voltage、 temperature, and switching activities, and are highly design dependent [6]. The other kind of sources are physical factors which arise due to integrated circuit manufacturing process and wear-out (electromigration), and usually permanent. These factors refer to the variations in characteristics of devices and wires, for example, effective channel length, gate oxide thickness, threshold voltage, and so on. Physical factors can be further divided into two categories, Die-to-Die physical variations and Within-Die physical variations, respectively. The former ones are usually predict by using worst-case corners and are independent of the design implementation. The later ones are Within-Die physical variations, compared with Die-to-Die variations, they depend on the circuit layout and there is no general method yet exist to predict them [6].

From another perspective, the parametric variation components can be divided into four categories. The first one is global variation between different chips, and it is given by case files during the designing. This may include fab, lot, wafer, and reticle level variations. The second kind of variation component are local variations, it comes from totally random variation, and mainly influences the performance of differential circuit, such as current mirrors and differential amplifier offset voltage. Then the third category is position dependent variations, this is resulted from across chip gradients. And it plays an important role in global clocks, large analog blocks, and long critical paths. The fourth kind of variation is based on proximity and orientation, and it is not random but systematic and deterministic. This kind of variations can be improved by process changes, and can also be modeled and simulated [7].

Due to such variability and margins for devices and interconnect are not scaling at the same rate, those sources of variability aroused a larger concern. Figure 3.1 illustrate the device and interconnect variations trends for different technology generations. As we can see from Figure 2.1, the percentage of total variation of some device parameters increased gradually from 1997 to 2007. During that period of time, the wire geometry, such as width W, height H, thickness T, and resistivity ρ represents a significant increase. The ratio of effective channel length also goes up from 40% to 65%. The parameters such as gate oxide thickness and threshold voltage increase slightly [8]. As a result, methodologies and models which can deal with such variation trends will play an increasingly essential role in circuit design.

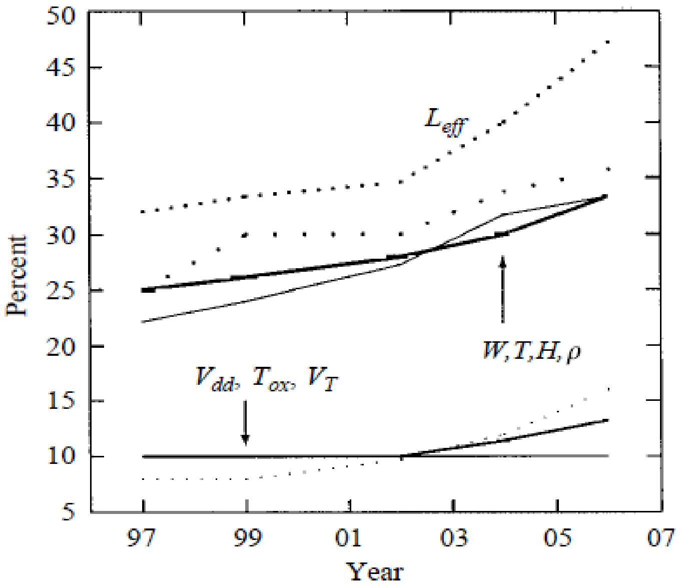


Fig 3. 1 Device and interconnect variation trends for different technology generations.

(Figure from [8])

3.1.2 Solution of the variation trend

To deal with such variation trend, the variation-tolerant design is required. This means if there is process variability, we can change the way we design to make the integrated circuit has a better response. Making the circuit works at a high clock frequency and has low power consumption as well. This likes an interesting trade off. There is no free lunch, you win and at the same time you lose. For example, as Figure 3.2 shows, we can make the transistor size larger, such as increasing the value of transistor length, this may make the device circuit less sensitive to process variation. In this way, we can get a better probability in frequency, but we need to pay it by higher power consumption and size, so it is a cost factor. Hence, we need to balance power consumption and frequency by using variation tolerance design. The approaches to achieve the variation tolerance design include worst-case design, careful choice of logic styles, self-adapting design, etc [7].

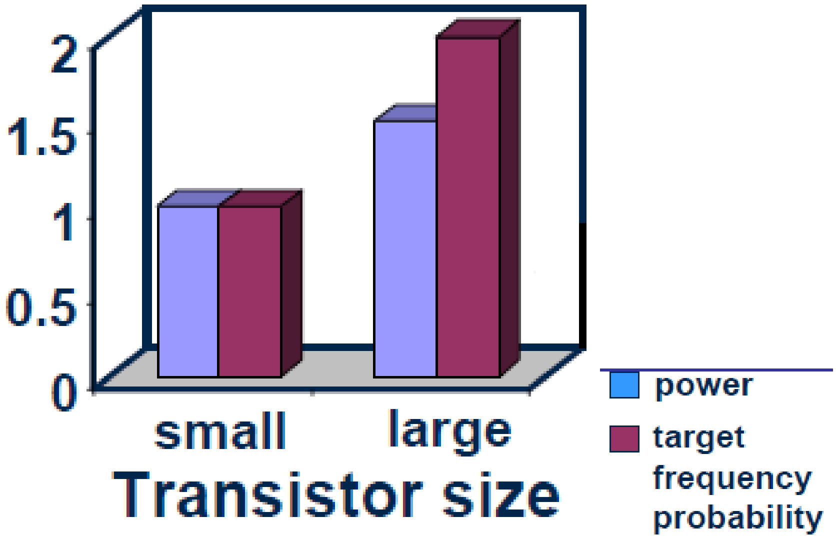


Fig 3. 2 frequency probability and power consumption in different transistor size

(Figure from [7])

3.2 SRAM cell

Embedded memory is one kind of integrated on-chip memory which can support the core on chip to accomplish particular functions and communication between cores. With its high performance and wide bus-width capability, embedded memory becomes an essential component in VLSI. Hence, in this project, we will analyze the impact of process variation on memory cell performance, particularly on SRAM cell stability. There are different kinds of embedded memory, for example, DRAM, FRAM, MRAM, PCRAM, FLASH, and SRAM [7]. Recently, many applications are using embedded memory, particularly present a high interest in SRAM.

SRAM refers to static random access memory. Due to its general and critical feature of memory, SRAM yield is very worth to study. As density is a very important factor for memory, SRAM uses the smallest device sizes compared with other kinds of memories [9]. Because the random dopant fluctuation can result in the threshold voltage variability, the gate area can be influenced. Due to this point, the smaller sized devices being used in a memory, the more impacts of random dopant fluctuation. The complex read and write requirement of SRAM cells need to be satisfied by a careful design, so that SRAM can perform a stable operation [9]. However, the parameter variability can cause the memory cell various. With the sensitivity of SRAM yield to process variation is increasing, a statistical analysis of PVT variation effects using Monte Carlo simulation is required.

3.2.1 Classical 6T SRAM

The classical design of SRAM memory cell usually uses six-transistor CMOS architecture. This is due to good electrical features of that architecture, for instance, power consumption, noise immunity, speed, and so on. Here, we proposed a 6T SRAM to illustrate basic structure, working principle, and the operations of SRAM. The schematic of 6T SRAM shows in Figure 3.3.

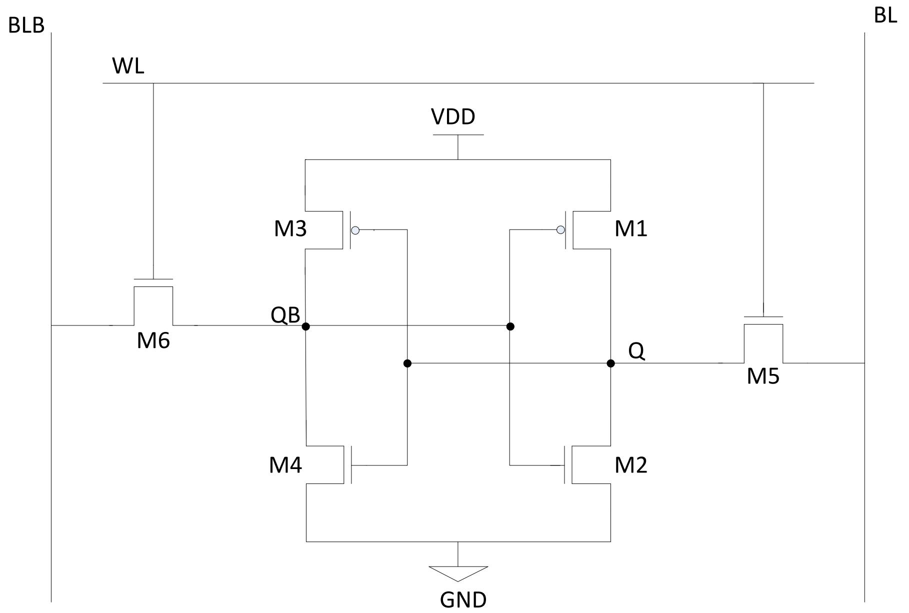


Fig 3. 3 The schematic of a 6-T SRAM cell

3.2.2 Working principle of 6T SRAM

In a six-transistor CMOS SRAM cell, there are four transistors formed as two cross-coupled inverters which are used for store each bit of data. These two cross-coupled inverters have two states which can be used to indicate 0 and 1. Another two transistors are used to control the access to the storage cell which refers to those two cross-coupled inverters when read and write operation happens. A SRAM generally uses six transistors to store data, sometimes other kinds of SRAM such as 8T, 10T also used to accomplish more than one port.

The word line (WL in Figure 2.3) can enable the access to the SRAM cells, it controls the two access transistors M5 and M6 to decide when the cell can be connected to the bit lines (BLB and BL). Both read and write operations will use those two bit lines to transfer data [10].

3.2.3 Three operation states of 6T SRAM

There are three different states of the SRAM cell, which include standby when the circuit is free, reading if the stored data need to perform reading, and writing when the storage cell requested to write the new contents [10].

Standby state is when the word line is not enabled, those two access transistors M5 and M6 are not connect to the bit lines. In this situation, transistors M1 to M3 which refers to those two cross-coupled inverters will continue to enhance each other due to the connecting to the supply voltage.

When the SRAM cell is requested to perform reading, we can consider the reading content of the memory is 0, stored at Q, and 1 is stored at QB. In the beginning of the read cycle, both of the bit lines should be precharged to a logical 1, then enable the word line to enable the access transistors M5 and M6. The second stage is when the data which should be stored in Q and QB now need to be transferred to the bit lines. As we can see from the figure3.4, because both QB and BLB are one, then there is nothing happens between these two points. For the right hand side, BL is precharged to 1 and Q is 0 now, so the capacitance on the right hand side would be discharged. If the reading content is a 1, then the operation should be opposite, this means BLB will be pulled tend to 0 by discharging the capacitance on the left hand side. Then these two bit lines will present a small difference, and they reach a sense amplifier. As the sensitivity of sense amplifier is higher, the speed of the read operation will be faster.

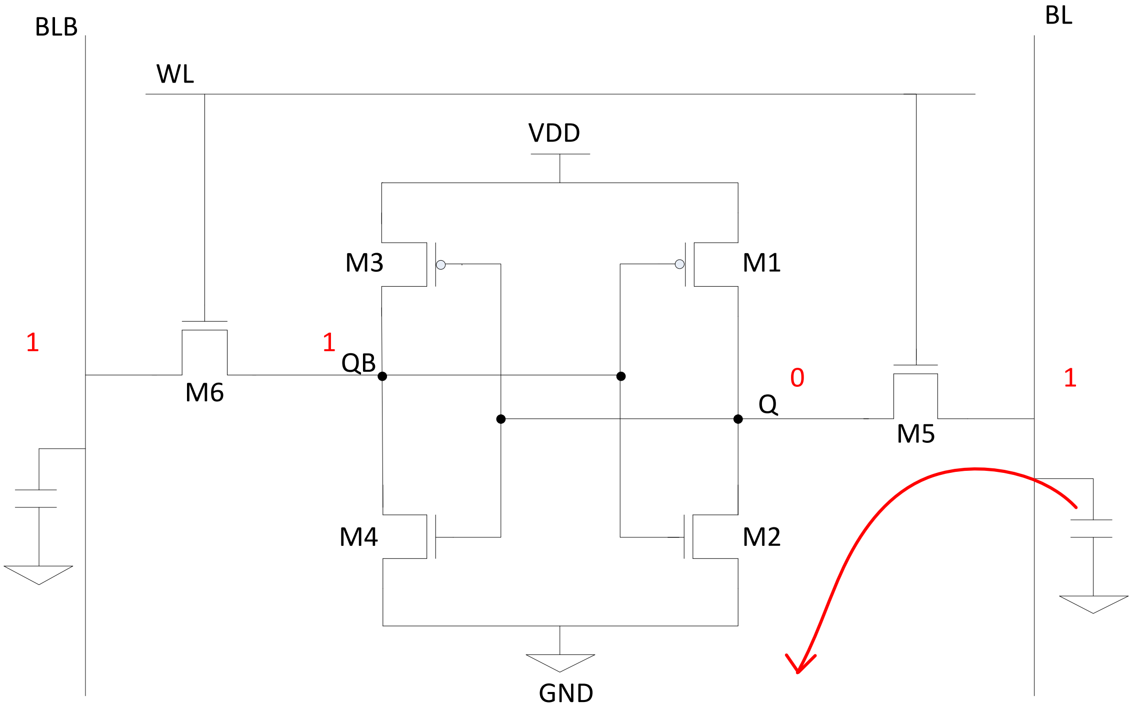


Fig 3. 4 The read operation of a 6-T SRAM cell (read 0)

In the beginning of write cycle, a value should be applied to be stored to the bit lines by using the write driver. If it is a 0, we would apply a 0 to the bit lines. For example, making BLB to 1 and discharging BL to 0. When we wish to write a 1, this will be accomplished by inverting the logical values on the bit lines. At the same time, the word line should be enabled to control the access of the data.

3.3 Statistical analysis

This project proposed the statistical analysis to investigate the influence of process-voltage-temperature variations to the performance of integrated circuit. Although there are many approaches utilized to analysis and address process variability, most of these methodologies are based on statistical modeling and mathematical optimization methods. This is because statistical analysis not only illustrates the deviation from expected circuit performance, but also analyzes the deviation from intended parameter variations.

3.3.1 Corner analysis

In this project, various process corners would be used to study delay and power variation of particular circuit by using Monte Carlo simulations. Corners can be considered as obvious natures of the objective. Process corners can be simply defined as a vector of extreme values of all parameters [11].

Traditional corner analysis refers to checking all process corners which come from the extreme values of all parameters by running static timing analysis (STA). It is attempts to assure that all the corners meet the circuit timing constraints. As the number of process corners is increasing, the traditional corner analysis is becoming expensive [11]. After 130nm technology library node coming up, the circuit design based on three kinds of corners, which are Typical, Worst, and Best corners. The typical corners based on both worst and best extreme corners, and they generally used to present the power dissipation under nominal conditions. Worst corners associated with high temperature, low supply voltage, and slow-slow process, they could result in worst timing delay. The Best corners associated with low temperature, high supply voltage, and Fast-Fast process to reach a perfect performance [12].

As the cost of the traditional corner analysis is increasing, an alternative method can be used which is linear-time method for STA. This approach can cover all process corners in a single pass which can present a more accurate result [11].

3.3.2 Statistical model vs. corner model

In statistical model, the parameter variations can be defined in terms of statistical distributions. It is not hard to see that statistical model can be considered as the sum of the various corner cases. On the other words, the corner models seems to be the specific example of the statistical model, as there are extreme values in the statistical model. With the high requirements of the circuit design, designers can not only rely on the corner models to produce a better design [1]. In this situation, the statistical models are needed by creating the large number of Monte Carlo simulations.

3.3.3 NMOS variation and PMOS variation determination

As explained in [13], a Gaussian distribution is safe assumption for statistical analysis. We can consider that the process parameters follow a Gaussian distribution, so that the NMOS and PMOS variation can be evaluated by knowing the values of extreme process parameter. We can directly get these values from the vendor’s corner model libraries. If sometimes these values are not given, they can be evaluated by using the silicon vendor’s process documentation. Considering the parameter of device models as Pj, then the distribution of it is:

$$\mathrm{Pj} = \mathrm{agauss}\ \left(\mathrm{P_{jTT}}, \frac{1}{2}\left(\mathrm{P_{jFF}} - \ \mathrm{P_{jSS}}\right),\ \ \sigma\right), \forall \mathrm{j} \qquad (1)$$

Where,$\mathrm{P_{jTT}}$, $\mathrm{P_{jFF}}$ and $\mathrm{P_{jSS}}$ are the values of parameter for the typical, fast, and slow libraries, and σ is the sigma difference between the fast and slow libraries. As the correlation of NMOS and PMOS is very important, it cannot be ignored. We need to add this factor, so that the equation of creating the statistical models is more accurate. Correlation in the statistical model evaluated by using the value of parameter in this form:

$$\mathrm{P_j} = \sqrt{\mathrm{c}} * \mathrm{P_{jc}} + \sqrt{1-\mathrm{c}} * \mathrm{P_{ju}}, \forall \qquad (2)$$

Where c can be considered as the correlation factor, P_{jc} is the correlated components and P_{ju} is the uncorrelated components of P_j. And both of them can take in equation (1). Equation (2) follows the equation:

$$\sigma_{\mathrm{Pj}}{}^2 \equiv \sigma_{\mathrm{Pjc}}{}^2 + \sigma_{\mathrm{Pju}}{}^2$$

This means, both correlated and uncorrelated components result in $\mathrm{P_j}$. By combining equation (2) and (1), the final form of can be created as follows:

$$\mathrm{P_j} = \mathrm{P_{jTT}} + \frac{1}{2}\left(\mathrm{P_{jFF}} - \mathrm{P_{jSS}}\right)\left(\sqrt{\mathrm{C}} * \mathrm{aguass}\ (0, 1, \sigma) + \sqrt{1-\mathrm{c}} * \mathrm{agauss}(0,1,\sigma)\right), \forall \mathrm{j}$$

This equation can be used to create a statistical device model.

3.3.4 Key elements of statistical circuit analysis

3.3.4.1 Extraction of statistical device models

To create a statistical device model, the designers need to extract the model parameters which can perform the circuit simulation. For example, there is a device model which refers to the Spice Level-1 MOSFET model. The parameters of this model can be extracted from the measurements of current, charge, and voltage. By measuring many devices, we can characterize the variability in those parameters have extracted, hence, there will be large amount of cases can be simulated. After analyzing the result of those simulations, the designer may clearly see the effects of the parameter variations [8].

However, there are some limitations for this method, such as high computational costs, large number of data collection, and potential errors during performance estimating. Thereby, the worst-case analysis methods or many other analysis techniques should be used to get the reliable and accurate statistics.

3.3.4.2 Sensitivity analysis

We can use sensitivity analysis to analyze the influences of the parameter variations. In some situations, it is easier to extract sets of the parameters than predict the effects to the device model. In those situations, we can consider that the variability propagate from one parameter P to another parameter Q, and Q is a function of P, where Q= f(P) [8].

In order to evaluate the value of f, Monte Carlo or other sampling methodologies are often used. Sensitivity analysis also can be used through a first-order expansion of function f :

$$Q + \Delta Q = f(P + \Delta P)$$

Where, $\Delta\mathrm{Q} \approx \left|\frac{\partial \mathrm{f}}{\partial \mathrm{f}}\right| \Delta\mathrm{P}$, and ΔP and ΔQ are the standard deviations of parameters P and Q. ΔQ can be evaluated by a normal distribution. As result, the impact of the parameter P variations on device models can be derived [8].

3.3.4.3 Worst-case analysis

The different of components of P are sometimes correlated. That correlation of P is needed to make an accurate statistical analysis. The worst-case analysis is the most common approach for evaluating the implications of correlated and random variations. Consider a device model performance is z, and it is a function of model parameters P, refers to z = f (P) [8]. Because of the variability of parameters P, the performance of the device model is variable. To evaluate the quality of the design is the goal of worst-case analysis. The yield of the design can be considered as the measure, it is defined as the percentage of the integrated circuits that meet the specifications made by the designer. As simulating such large amount of the circuits is very expensive, worst-case analysis is a better method to measure the yield of the design.

3.3.4.4 Spatial modeling and mismatch analyses

Device matching plays an essential role in analog circuit design, during this process, a series of work has been done to analyze and model issues. Device matching also arouse a concern in high performance digital circuits.

According to [8], consider ΔP is the variance in some parameters between two devices. In ideal case, if ΔP =0, it means it is a perfect matching. However, we cannot always achieve such perfect matching, and this situation produce a sense of possible mismatch. And we need to take measures to reduce this mismatch.

One situation is if the two devices are totally independent, then the variation of mismatch is twice variation of each device. Another case is if any of the device variation resulted from the device area, and when the potential source of variability can be “averaged” out for the larger devices, then the variation of mismatch based on both the potential statistics and the size of each device. In this case, it may reach a better matching by paying the cost of making the circuit structure larger.

The third situation is the mismatch resulted from the distance between two devices. A helpful evaluation for this is to make the variation of the mismatch to be proportional to the square of the distance of the two devices. The helpful method to get a better matching under this situation is making the two devices on the die as closer as possible.

A final case is if there is some deterministic spatial process variability. In the early stage, this kind of systematic variability must be considered as random. Consider a parameter P, its variations are systematic and the total variations can be formed follow this equation:

$$\mathrm{P} = \mathrm{P}_0 + \mathrm{F}(\mathrm{x}, \mathrm{y}, \theta) + \widetilde{\mathrm{P}_\epsilon}$$

Where P_0 is the mean value of the device, function F express the device spatial distribution of P, θdefined as a vector of random parameters of that function. We can consider $\widetilde{\mathrm{P}_\epsilon}$ to the unexplained random parameter variation. If the information of (x, y) is unavailable, then the parameter P can only be considered as a random distribution which has the variation associated with the maximum deviation resulted from F. One method is to consider x and y have the random distribution across the die, then use such distribution to evaluate the function F. Generally, this approach will increase the total variation of P.

3.4 Circuit level simulation using Monte Carlo

Monte Carlo is a computational method which based on repeated various sampling to evaluate their results. It is usually used in simulating analysis and mathematical systems. In the integrated circuit simulation using Monte Carlo method, our goal is to simulate various parameters simultaneously, and make statistical analysis for the variability of the performance of the circuit [14]. This performance can refers to quantifying the statistical variation of circuit delay and power. Matlab is matrix laboratory, it is a mathematical computing environment which can plot the functions and various data [15]. Hspice is usually used for accurate simulation for integrated circuits, and it is an effective tool for simulation.

According to Figure 3.5, the procedure of Monte Carlo simulation is firstly choose a particular circuit to study. Secondly, extracting all parameter values randomly use respective Gaussian distributions. All parameters can be considered independent to each other. In the third stage, parameters can be applied to circuit to produce a particular instance. Fourthly, simulate those instances using HSPICE to measure the circuit output parameters. The read time and write time are proposed as the output parameters of the SRAM cell in this project. Finally, go back to the second step and repeat it again and again for N times. Then Matlab can be used to show all the results in a graph, so that we can easily see how the value of the read time and write time changes.

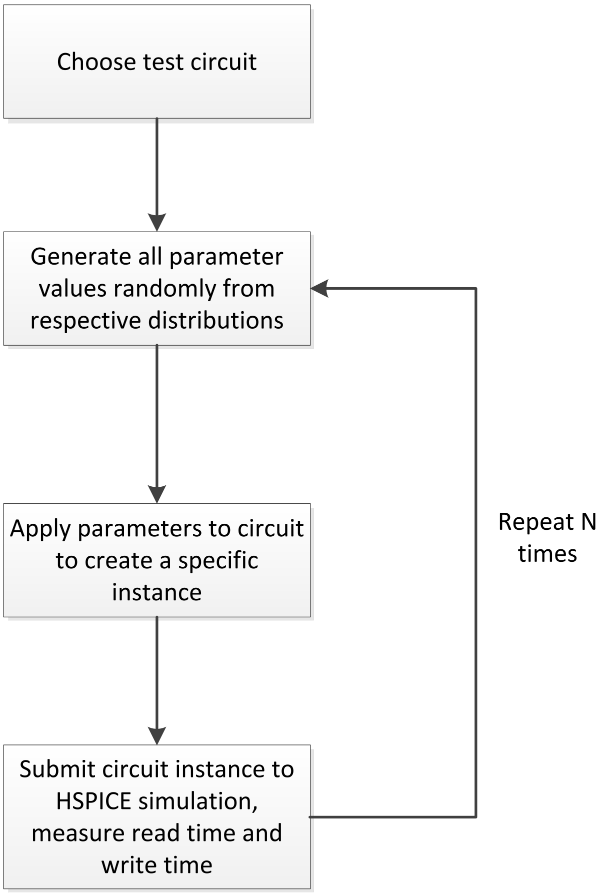

Choose test circuit
Generate all parameter values randomly from respective distributions
Apply parameters to circuit to create a specific instance
Submit circuit instance to HSPICE simulation, measure read time and write time
Repeat N times

Fig 3. 5 The procedure of Monte Carlo simulation

3.5 Statistical Blockade

For the circuits in nanoscale, process-induced random variations are dramatically increased in devices. Such variations may include random dopant fluctuations, line edge roughness, and gate oxide variation. This roused a growing concern in the field of circuit reliability with statistical process variation. As circuits like SRAM and flip flops are high replicated, there are millions of cells in one chip. We can consider the cell failure as the rare event, in this situation, a rare statistical event could easily cause the whole system failure. As we can see from the Figure3.6 [16],

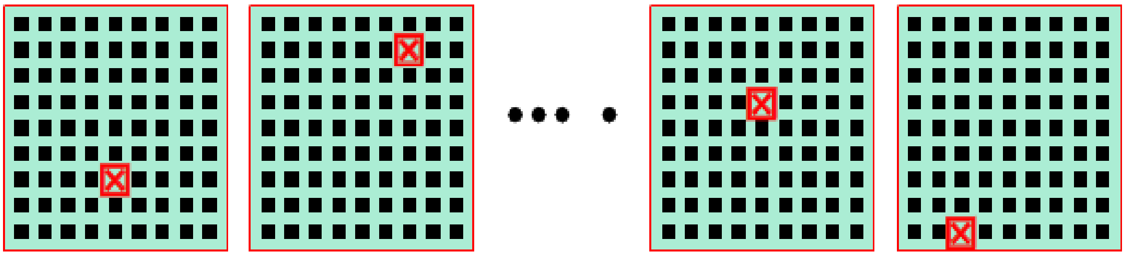


Fig 3. 6 Millions of memory cells on the chip [16]

On the other hand, the existing techniques which are used to test and find the rare events are not performing so well. For example, standard Monte Carlo is an efficient method to sampling and mathematical systems. It can be used to detect the statistical rare events, however, for such high replicated circuits like SRAM, there should be a large amount of simulation, so it will cost of time when sampling the rare circuit events. As we need to generate a large number of simulations if we use standard Monte Carlo simulation method, the experiment would be much expensive although such method is accurate. Compared with standard Monte Carlo simulation, statistical blockade mainly focus the Monte Carlo to the rare events. So the generating by statistical blockade is much cheaper than standard Monte Carlo simulation [3].

Hence, how to propose a fast and accurate method for generating an accurate prediction of the rare circuit event under PVT variation is important. In this situation, the statistical blockade (SB) method is increasingly concerned. In this project, we will propose to consider statistical blockade approach first followed by other techniques.

3.5.1 The basic idea of calling Statistical Blockade

As we analyzed before, the standard Monte Carlo simulation is too slow to model the rare events. Therefore we need to speed up the simulation of the critical events. If we can focus the simulation only on the rare events, it should be faster. The rare events can be defined as the tail part of the distribution. As analyzed in [17], the Importance Sampling might help us to sample the tail points of the distribution [17], however, it will change the statistics of these tail points. Thus, we also need to raise reliability of the statistics.

In this project, we proposed the Statistical Blockade method to do a very efficient and novel simulation on the rare circuit events. Firstly, we can only focus on the critical events so that the simulation samples can be reduced. Secondly, unlike the Importance Sampling, Statistical Blockade would not change the statistics of the rare events. This may increase the reliability of the evaluation of the rare circuit events. Thirdly, this technique use mathematics of Extreme Value Theory [18] to model the tail part of the distribution, so that we can build the classifier to efficiently filter the samples and block the points which are unlikely to fail.

3.5.1.1 Use the Extreme Value Theory to model the rare events

EVT help us use a mathematical method to build the model of the tail part of the distribution [16]. Consider that t is the threshold of the cumulative distribution (CDF) F(x). This means the tail part of the distribution is consisted of the values above t. Through this project, we consider the upper tail as the probability to fail in circuit production. Then the conditional CDF which exceed the threshold t can be defined as below:

$$F_t(x) = \frac{F(x+t) - F(t)}{1 - F(t)} \; for\ x > 0$$

Now we want a similar result of the tail part, and this can be done by the important distribution of the Extreme Value Theory, which refers to Generalized Pareto distribution. It can be defined as follows:

$$G_{a,k}(z) = \begin{cases} 1 - \left(1 + \frac{kz}{a}\right)^{1/k} & , k \neq 0 \\ 1 - e^{-z/a} & , k = 0 \end{cases}$$

Then we can fit GPD to the conditional CDF.

3.5.1.2 Fitting the model and prediction

We can fit the GPD by evaluating the value of a and k, so that we can fit GPD to a conditional CDF. There are several options available to achieve this idea, such as moment matching, maximum likelihood estimation(MLE) and probability weighted moments (PWM) [27]. We choose PWM in this project because it seems to have lower bias [19] , moreover it does not have the convergence problems of MLE.

3.5.2 The implementation of the Statistical Blockade

The idea of Statistical Blockade method also can be shown in Figure3.7 [3]. As we can see from the graph, generating the large amount of the Monte Carlo samples is easy and cheap, however, simulation is costly.

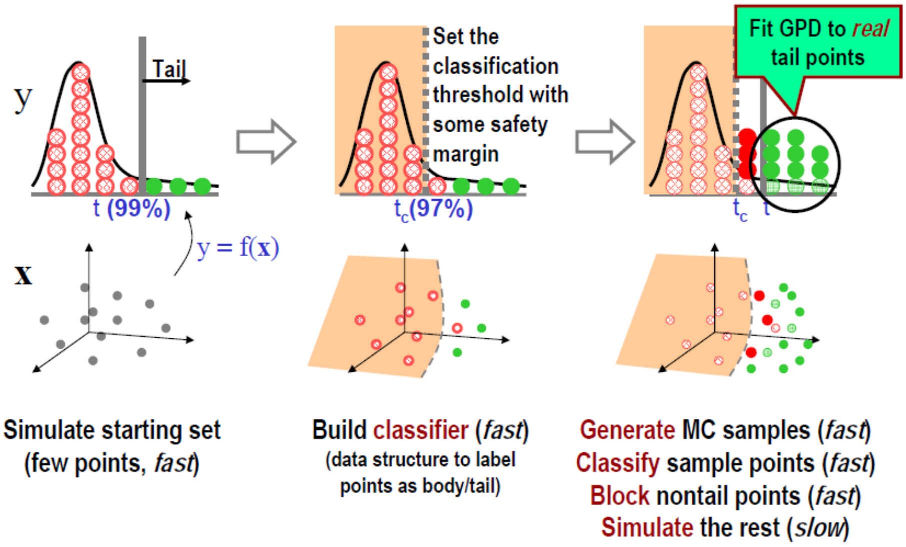

y
Tail
t (99%)
y = f(x)
x
Set the classification threshold with some safety margin
t_c(97%)
Fit GPD to real tail points
t_c
t
Simulate starting set
(few points, fast)
Build classifier (fast)
(data structure to label points as body/tail)
Generate MC samples (fast)
Classify sample points (fast)
Block nontail points (fast)
Simulate the rest (slow)

Fig 3. 7 The idea of Statistical Blockade (Figure from [3])

3.6 Low-leakage SRAM design with dual Vt transistors

3.6.1 The power consumption of a circuit

The power consumption of a circuit can be divided into two categorize, one is dynamic power, and the other one is static power. Previously, dynamic power is the main power dissipation of the whole circuit. However, recently the static power is increasingly important in the total power consumption [20]. As the technology scaling, how to reduce the increasing static power is one of the most important problem in the design of VLSI systems.

Static power is produced by many reasons, for instance, subthreshold leakage, reverse-biased P_N junctions, drain-induced barrier lowering (DIBL), gate-induced drain leakage, punch through currents gate oxide tunneling, and hot carrier effects [20]. And the subthreshold leakage is the dominant element of the static power. Hence, static power can be defined as the product of supply voltage and the leakage current which expressed in the equation below:

$$P_{static} = I_{leakage}V_{DD}$$

Leakage power is the power dissipated by the transistors which are not active. Therefore, the transistors which are not active should be found, and then we need to calculate the leakage power for each of them and add them together.

3.6.1 The idea of dual threshold voltage design

Previously, we want to reduce the dynamic power consumption, hence, the supply voltage should be decreased. This would be achieved by reducing the threshold voltage of the transistor. According to the equation below, when the threshold voltage is decreased, the subthreshlod current will be increased. The leakage power will be increase as well.

$$I_{subthreshold} = \mu_0 C_{ox} K \frac{W}{L} (1 - e^{\frac{V_{DS}}{V_t}}) e^{\frac{(V_{GS} - V_{th})}{nV_t}}$$

Where, μ_0 is the movement speed of the carrier, C_{ox} is the capacitance of the oxide layer, W and L are the width and length of the CMOS transistor, K and V_t are the thermal approximation constant and the thermal voltage, V_{th} is the threshold voltage of the transistor.

As we analyzed above, recently the leakage power is increasing dramatically, thus, how to reduce the leakage power is of growing concern. Previously, some researchers have solved this problem. The asymmetric SRAM cells were used to reduce the leakage power [21]. The circuit level techniques which add the device-level optimization were also proposed to provide a forward body-biasing method. This method could reduce the leakage power in hold and active status [22]. And the technique of the dynamic V_t has also been carried out to decrease the leakage power of the SRAM memory cells [23]. However, although those methods could reduce the leakage power of SRAM cells, most of them would cause the increase in the area of the chip and the decrease in the yield of the circuits.

In this project, we proposed a method of low-leakage SRAM design by using three different types of the memory cells, which is created by various threshold voltage assignments. This is because the leakage power could be reduced by using a high threshold voltage for some transistors. We consider all types of memory cells are symmetric.

There are four main merits of using dual threshold voltage design [24] compared with previous method. Firstly, there is no hardware overhead at all. Secondly, the read time and write time would not overhead as well. Thirdly, the method would not change the SRAM design a lot. Fourthly, it could make an improvement on the static noise margin under process variation.

3.7 Static Noise Margin

3.7.2 The concept of static noise margin

The cell stability is one of the most important aspects of the SRAM cell. It influences the soft-error rate and the sensitivity of the memory to process tolerances and operating conditions. In this project, the Static Noise Margin of 6T SRAM under process variation was measured with all its status [24].

Static Noise Margin (SNM) can be defined as the biggest value of the noise voltage allowed between the two inverters in the SRAM memory cell, and with this value the memory cell can still maintain its states [25]. We can define the noise voltage as V_n. The Static Noise Margin of the SRAM can be considered as the maximum value of the V_n that can be added at the point "Q" and "QB" without changing the data stored in the memory cell. As shown in Figure3.8 [24], he estimating structure of the measuring the SNM can be presented like the flip-flop.

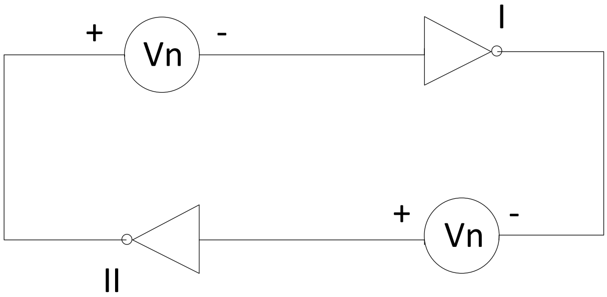


Fig 3. 8 A flip-flop comprised of two inverters. Static-noise voltage sources V, are included.

3.7.3 The calculation of Static Noise Margin

As we know the structure of the SRAM cell, there are two cross-coupled inverters in the memory cell. A graphical method of evaluating the Static Noise Margin is by drawing the characteristics of one inverter and mirroring it. Then find the maximum possible square between those two characteristics. The original and mirrored inverter characteristics are shown in Figure3.9 [26], they can be considered as the two functions $y = F_1(x)$ and $y = F_2(x)$. The distance between the two curves is the diagonal of the square, multiply the value of diagonal by cos45, then we can get the value of SNM.

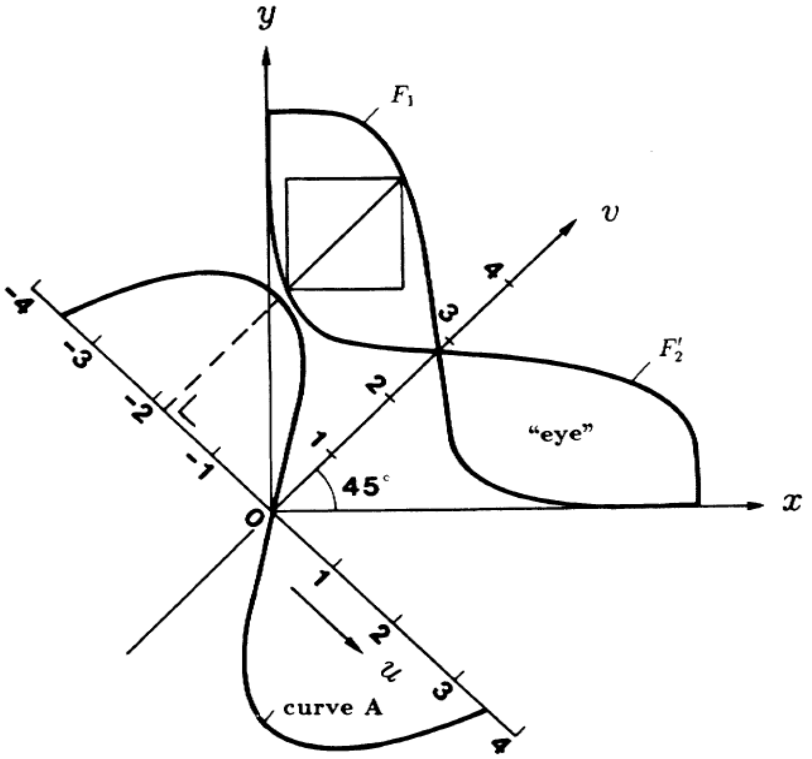


Fig 3. 9 Graphical representation of the Static Noise Margin (Figure from [26])

There is a procedure we need to take to estimate the diagonal of the maximum square. We need two coordinate systems to calculate it. One is the normal one, which refers to the (x, y) coordinate system. And the other one is (u, v) coordinate system. The later one is rotated through 45° with the normal coordinate system [26]. At given u values, the length of diagonal can be measured by the subtraction of the values of v in original and mirrored characteristics.

To get F_1 under (u, v) coordinate system, its value in (x, y) coordinate system need to be transferred to (u, v) coordinate system. Operator R [26] shown below can be used to finish rotating the (x, y) system, and transfer it to (u, v) system.

$$R\vec{x} = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \vec{x} = \vec{u}$$

Where $\vec{x}$ and $\vec{u}$ are the column vectors (x, y) and (u, v). x and y could be defined as the function of u and v as following expressions.

$$x = \frac{1}{\sqrt{2}} u + \frac{1}{\sqrt{2}} v$$

$$y = -\frac{1}{\sqrt{2}}u + \frac{1}{\sqrt{2}}v$$

If we substitute the function F_1, we could get the following expression:

$$v = \sqrt{2}F_1\left(\frac{1}{\sqrt{2}}u + \frac{1}{\sqrt{2}}v\right) + u$$

To get F_2, F_1 should be mirrored in the system of (x, y), and respect to $y = x$ at the same time. The matrix form of the mirroring is as follows:

$$M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

So that we can get the next formulation showed below:

$$RM\vec{x} = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\vec{x} = \vec{u}$$

which gives:

$$x = \frac{-1}{\sqrt{2}}u + \frac{1}{\sqrt{2}}v$$
$$y = \frac{1}{\sqrt{2}}u + \frac{1}{\sqrt{2}}v$$

If we substitute the function F_2, we can get following formulation:

$$v = \sqrt{2}F_2\left(\frac{-1}{\sqrt{2}}u + \frac{1}{\sqrt{2}}v\right) - u$$

Now the length of diagonal can be provided by the subtraction of the values of v in original and mirrored characteristics.

4. Methods

The methods and algorithm proposed in this project are explained in this section. It also presents the procedure for developing these methods.

4.1 Analysis using the Statistical Blockade method

4.1.1 The algorithm of Statistical Blockade method

The Statistical Blockade approach is the basic idea to optimize the analysis and simulation of the critical circuit events. Compared to standard Monte Carlo, it provides a better performance on the simulation time of the rare events.

Supposed that the size of training sample is n_0 (e.g. 1000), the size of total size is n(e.g. 10000), and the percentage P_t(e.g. 99%) and P_c(e.g. 97%). The Algorithm 1 [3] of the Statistical Blockade can be shown as follows:

Algorithm 1: For Statistical Blockade method

```
Assume: training sample size $\boldsymbol{n_0}$ (e.g., $\boldsymbol{n_0 = 1000}$); total sample size $\boldsymbol{N}$ (e.g.,$\boldsymbol{N = 10000}$)
$\boldsymbol{X = MonteCarlo\ (n_0)}$
$\boldsymbol{y = f_{sim}(X)}$
$\boldsymbol{t = Percentile(y, P_t)}$
$\boldsymbol{t_c = Percentile(y, P_c)}$
$\boldsymbol{C = BuildClassifier(X, y, t_c)}$
$\boldsymbol{y_{tail} = F_{sim}(Filter(C, N))}$
$\boldsymbol{y_{tail} = \{y_i \in y: y_i > t\}}$
$\boldsymbol{(\varepsilon, \beta) = FitGPD(y_{tail} - t)}$
```

Fig 4. 1 The algorithm of the Statistical Blockade method [3]

Initially, the simulation of a small amount of samples used HSPICE were carried out, which are the training samples for the classifier. And examine two thresholds, one is t which can be tested by the percentage P_t, and the other one is the threshold of classifier t_c which can provide a safety margin. And the classification threshold can be evaluated by the percentage $\mathrm{P_c}$. In the statistical parameter space, the n points can be generated using the function $\mathrm{MonteCarlo(n)}$. These points should be stored in a matrix, and the input dimensionality is the number of parameters we considered. Each point in the dimensions is regarded as a row of X. y is consisted of the output values of Monte Carlo simulations using HSPICE. Then we need to train and build the classifier by using the function $\mathrm{BuildClassifier}(\mathrm{X}, \mathrm{y}, \mathrm{t_c})$, where $\mathrm{t_c}$ is the classification threshold. After getting the classifier, the points in X are filtered, and the "body" points are blocked, then only return the "tail" points of the distribution. Now we need to fit the GPD to conditional CDF, this is done by evaluate the parameters ε and β by using the function $\mathrm{FitGPD}(\mathrm{y_{tail}} - \mathrm{t})$. Then the GPD model can be used to evaluate the statistical metrics for rare circuit events.

The Support Vector Machine (SVM) [27] are used as the classifier in this project, which is implemented as: LIBSVM [28], SVMlight [29] or WEKA [30]. There are 4 basic kernels of SVM , linear, polynomial, radial basis function (RBF), and sigmoid. Associated with the purpose of this project, the kernel RBF was chosen as it can nonlinearly map samples into a higher dimensional space.

Note that the classifier itself is body biased, the number of body points is much more than that of tail points. It will block the body points (nontail points) according to the classification threshold and the tail threshold, and keep the classifier try to find the body points as many as possible no matter if there is any tail point misclassified. Hence, the detected tail points would be misclassified. However, it caused another problem of how to reduce the misclassifying. In order to solve this problem, a method of weight scale to increase the weight of the tail points was proposed. By setting the value of the weight scale of the tail points, an appropriate number of the classified tail points can be obtained which make a safer margin than just use the classification threshold. For instance, if we just use the classification threshold which refers to 97%, there should be 300 tail points detected. On the other hand, it detected 349 tail points while using the technique of weight scale.

4.1.2 The algorithm of finding the minimal classification threshold

After accomplishing the method of Statistical Blockade, we consider that if we can further speed up the simulation of the rare circuit events. For given tail threshold, consider that we can find the minimal classification threshold and minimal training samples of the classifier which can still ensure to cover all the true tail points. If we could minimize the classification threshold and the training samples by this way, the total simulation would be reduced. Therefore, we could make the simulation of the rare circuit events faster.

To achieve this idea, the minimal classification threshold is required to be chosen. Then according to the given tail threshold and classification threshold, we could find the minimal training samples. Refer to figure4.2, an algorithm of setting the minimal classification threshold [31] was listed.

Algorithm 2 Choosing the minimal classification threshold $\mathbf{t_c}$ for given tail threshold $\mathbf{t}$

```
1:   Assume: training sample size $\boldsymbol{n_0}$ (e.g., $\boldsymbol{n_0 = 2000}$); total sample size $\boldsymbol{N}$ (e.g., $\boldsymbol{N = 10000}$)
2:   $\boldsymbol{Y = F_{sim}(N)}$
3:   $\boldsymbol{V = Percentile(Y, t)}$
4:   $\boldsymbol{Y_{tail}} = \{\boldsymbol{Y_i} \in \mathrm{Y}: \boldsymbol{Y_i > V}\}$
5:   $\boldsymbol{S = Size(Y_{tail})}$
6:   for all $\boldsymbol{t_c = t}$ do
7:       $\boldsymbol{x = MoneCarlo(n_0)}$
8:       $\boldsymbol{y = F_{sim}(x)}$
9:       $\boldsymbol{V_c = Percentile(y, t_c)}$
10:      $\boldsymbol{C = BuildClassifier(x, y, V_c)}$ //C is a classifier
11:      $\boldsymbol{y_{tail} = F_{sim}(Filter(C, N))}$
12:      $\boldsymbol{y_{tail,true}} = \{\mathrm{y}_{\boldsymbol{i}} \in \boldsymbol{y_{tail}}: \boldsymbol{y_i > V}\}$
13:      $\boldsymbol{S_c = Size(y_{tail,true})}$
14:      if $\boldsymbol{S_c < S}$ then
15:          $\boldsymbol{t = t - 1}\%$
16:          Repeat
17:      else
18:          $\boldsymbol{t_c = t}$
19:      end if
20:  end for
```

Fig 4. 2 The algorithm of choosing the minimal classification threshold [31]

For this algorithm, there were 2000 training samples to train the classifier and there are 10000 total samples. Firstly, the tail points of the whole samples are required to be selected. All samples were simulated by implementing the function called $Y = F_{sim}(N)$. Refer the expression, Y is the simulation result, for example, the read time or write time of the SRAM cell. V is the vector which contains the true tail points by using the function $V = Percentile(Y, t)$. Function $Size(Y_{\boldsymbol{tail}})$ is used to estimate the number of the tail points. According to Figure 4.2, step 7 to step 12 are completed to accomplish the method of Statistical Blockade. y is the result from the simulation of training samples. Meanwhile, the SVM classifier is trained and returned from function $C = BuildClassifier(x, y, V_c)$.

As soon as the classifier was built, set classification threshold equal to tail threshold firstly, then classify and block the body points in N using the classifier. The high performance was achieved by focusing on the filtered tail points simulating in HSPICE. The evaluation was based on the comparison of the size of the current tail points $S_{\boldsymbol{c}}$ and the size of the true tail points S. If $\mathrm{S}_{\mathbf{c}} < \mathrm{S}$, it indicated that the current t_c cannot select all of the tail points. Hence, the value of the classification threshold t_c required to increase. For each time it increased by 1%. Then repeat all of the steps again and again, by modulating those parameters, the minimal value for t_c can be eventually detected while $\mathrm{S_c}$ is equal to S.

4.1.3 The algorithm of finding the minimal training samples

Now the minimal classification threshold is found and the tail threshold was given, the minimal number of training samples should be evaluated. Refer to Figure 3.4, it illustrates the algorithm of choosing the minimal number of training samples [31]. Compared to the previous algorithms, the number of training sample should be changed from 100 to 2000, with increasing 100 as intervals. Two conditions are required to select the minimal training sample. Firstly, the size of the current tail points $\mathrm{S_c}$ is desired to equal to the size of the true tail points S. Secondly, the number of training samples which is larger than the current one also need to equal to the true tail points S.

Algorithm 3 Choosing the minimal number of training samples for the fixed classification threshold $\boldsymbol{t_c}$ and given tail threshold $\boldsymbol{t}$.

```
Assume: Initial training sample size $n_0$ (e.g., $n_0 = 100$); total sample size $N$ (e.g.,, $N = 10000$).
$Y = F_{sim}(N)$
$V = Percentile(Y, t)$
$Y_{tail} = \{Y_i \in Y : Y_i > V\}$
$\mathbf{S} = \mathbf{Size}(Y_{\mathbf{tail}})$
for all ($i = 1$; $n_0 \leq 2000$; $i++$) do
    $x = MonteCarlo(n_0)$
    $y = F_{sim}(x)$
    $\mathrm{V}_c = Percentile(y, t_c)$
    $C = BuildClassifier(x, y, V_c)$  //C is a classifier
    $y_{tail} = F_{sim}(Filter(C, N))$
    $y_{tail,true} = \{y_i \in y_{\mathrm{tail}} : y_i > V\}$
    $S_c = Size(y_{tail,true})$
    $n_0 = n_0 + 100$
end for
find($S_i = S \ and \ S_{i+1} = S_{i+2} = \cdots = S_{20} = S$)
$n = i * 100$
```

Fig 4. 3 The algorithm for choosing the minimal training sample[31]

4.2 Different deviations of parameters

Since SRAM cell use very small transistors, the influences of process variation are the most important in SRAM design. The decrease of the transistor scale may cause the increase of the process variation [32]. For example, if the scale of the transistors of the SRAM is decrease, the variation of threshold voltage would be widened [32] like Figure4.4. The simplest way to model this increased variation is to make the Gaussian distribution with a higher standard deviation.

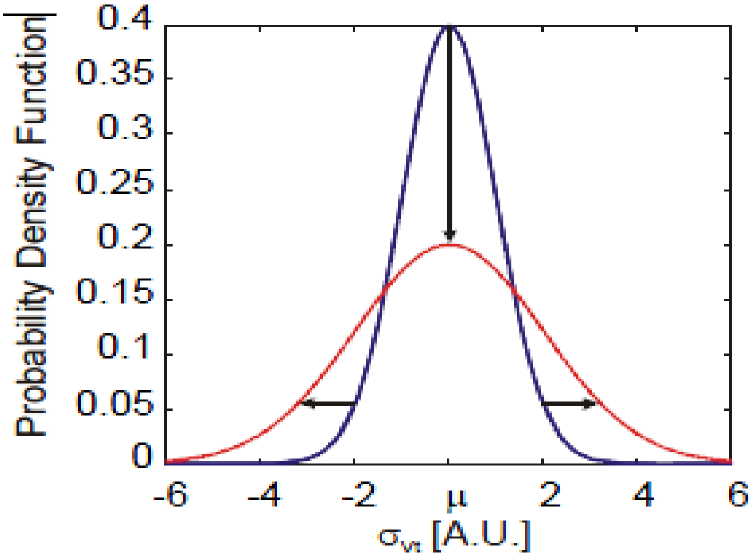


Fig 4. 4 The Gaussian distribution with higher deviation [32]

The deviation of the Gaussian distribution presents how much variation is from the mean value. If the points of the Gaussian distribution are very close to the mean value, then there is a low deviation. While, if there are a large range of the data points, the deviation should be high. Therefore, different deviations would result of different Gaussian distributions of parameters.

Refer to the discussion in introduction section, the Statistical Blockade method can be optimized by minimizing the classification threshold and the training samples of the classifier. The value of the selected classification threshold and the number of the training samples was unstable, while increasing the deviation of the Gaussian distribution. Hence, how the value of the selected classification threshold and the number of training samples would be changed in this situation is a question. The answer of this question could be answered by calling the method of the SB and the idea of minimizing the classification threshold and training samples of the classifier. The results of our simulation will present in section five.

4.3 Low leakage SRAM design

To design a low leakage SRAM cell, the first task to be completed is estimate the leakage power of the SRAM cell in each state. The active transistors need to be detected in terms of measuring the leakage current. Then use the leakage current multiply by the supply voltage, we can get the leakage power. Add the leakage power for each inactive transistor together to get the leakage power of the whole memory cell.

For instance, during the hold state, as we can see from the Figure 4.5, the transistor M1, M4, M6 are not active during this status. Thus, the leakage power (static power) of those three transistors should be measured. As we consider that word line is closed, bit line (BL) and bit line bar (BLB) were stably logical 1, and the data Q stored in the cell equalled to zero. The leakage power can be calculated based on the following equation,

$$P_{leakage} = \left(I_{leakage1} + I_{leakage4} + I_{leakage6}\right)VDD$$

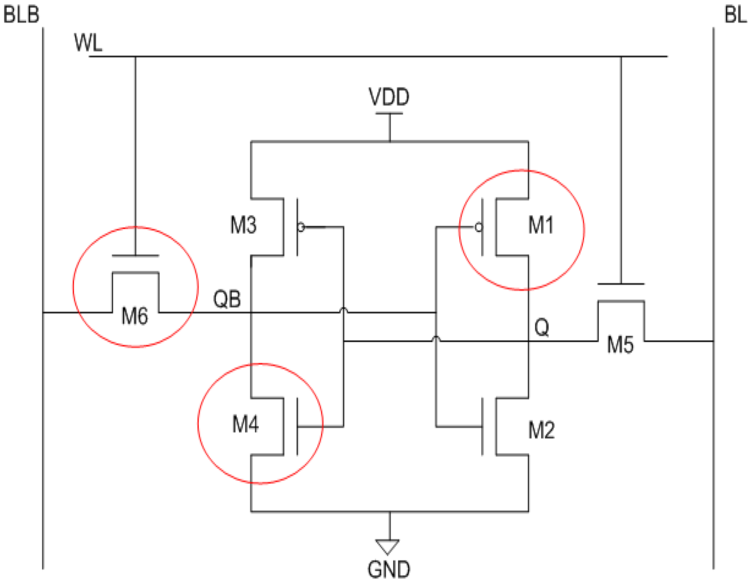


Fig 4. 5 The schematic of SRAM cell in hold state

When the SRAM cell asked to perform as read operation, the word line is enable which make the transistors M5 and M6 active. Suppose that the initial data stored in the SRAM cell is Q=0 and QB=1, this make the transistors M3 and M4 active. As a result, during the state of read, the transistors M1 and M4 are not active whose leakage powers need to be measured (Figure4.6). And the leakage power can be calculated as follows:

$$P_{leakage} = (I_{leakage1} + I_{leakage4})VDD$$

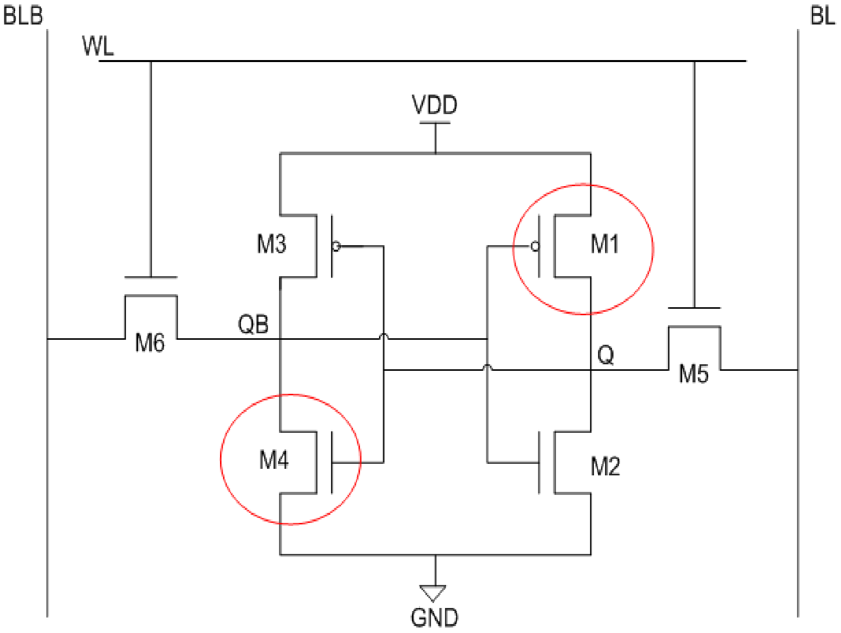
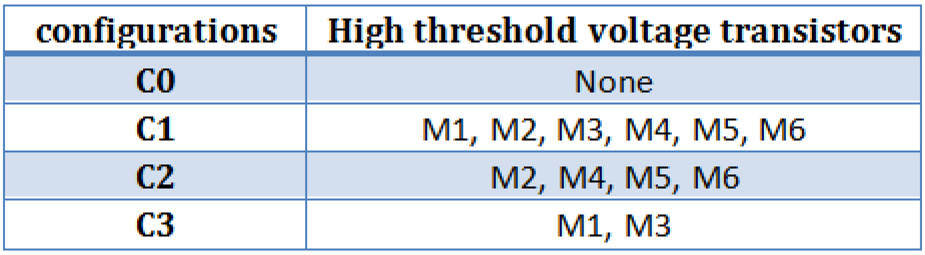


Fig 4. 6 The schematic of SRAM cell in read operation

When the SRAM cell is asked to perform a write operation, the word line is enable as well. And also set the initial data stored in the memory cell to Q=0 and QB=1. As result, transistors M2, M3, M5, M6 are active. The leakage power of the circuit is consisted of the leakage power dissipated by transistor M1 and M4. It is the same as the read operation.

Refer to the results obtained for read and write operation, the leakage power was relatively high. The design can be optimized by implementing a low leakage SRAM cell using dual threshold voltage transistor.

In order to reduce the leakage power of the SRAM cell, three kinds of symmetric cell configurations were proposed. This means symmetric CMOS transistors in a cell have the same threshold voltages. Associated with Table3.2, the configuration C0 had no high threshold voltage transistors, all of the transistors in configuration C1 have the high threshold voltage, four of the transistors in configuration C2 have the high threshold voltage transistors, and two of the transistors in configuration C3 have the high threshold voltage transistors.



configurations	High threshold voltage transistors
C0	None
C1	M1, M2, M3, M4, M5, M6
C2	M2, M4, M5, M6
C3	M1, M3

Table 1 The different configurations of low leakage design

For each configuration (except for C0), there are two values of threshold voltage transistors, one is normal value of threshold voltage, the other one is high threshold voltage. It carried out with a problem that there were two kinds of threshold voltage in one circuit. To fix this problem, two models are proposed in the simulation, and each model respect to each kind of threshold voltage. By varying the value of high threshold voltage from the normal value (0.46V) to 0.67V, the changes in the leakage current for different configurations were found.

4.4 Static Noise Margin

As we all know, there are three states of SRAM cell, standby, read, write respectively. The measurements of Static Noise Margin for those three states are slightly different.

When the SRAM cell is in standby state, the word line is not enabled, this means the access transistors M5 and M6 are closed. Hence, just the two cross-coupled inverters are holding the data, which is shown is Figure4.7. Then add a static noise voltage at the point Q and QB, and do DC analysis. We can get the butterfly in Figure4.8, and the high performance model with constant value of threshold voltage and gate oxide thickness were proposed. The supply voltage is V_{DD} = 1V. The SNM of the SRAM cell in this state can be defined as the side of the maximum square.

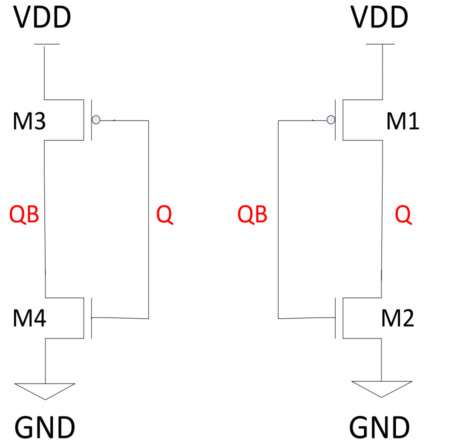


Fig 4.7 The schematic of SNM simulation in standby state

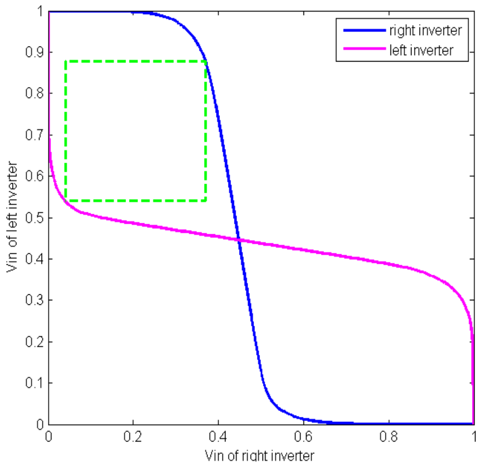


Fig 4.8 The SNM in standby state

If the SRAM cell is required to perform the read operation, the access transistors cannot be ignored any more (M5 and M6). The two bit lines need to be precharged to logical 1 to perform the read operation, this means the two bit lines can be connected to VDD. In this situation the SRAM circuit can be presented as Figure4.9. The static noise sources are also added to the points Q and QB to do the DC analysis. The two characteristics of the inverters are showed in Figure4.10.

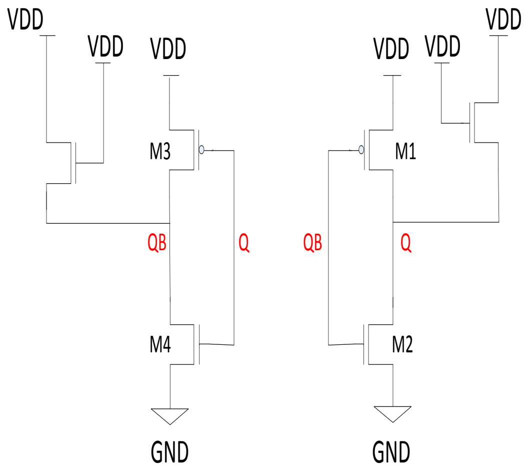


Fig 4. 9 The schematic of SNM simulation in read state

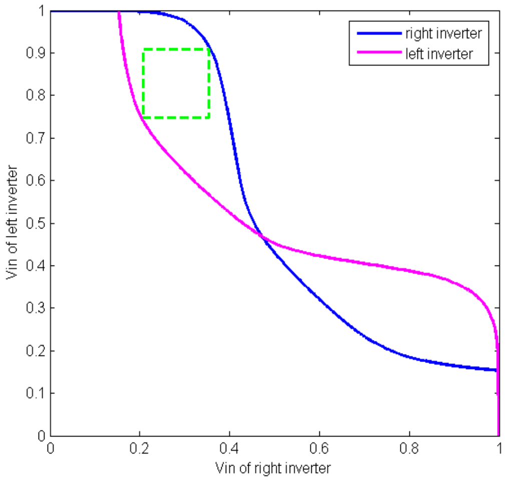


Fig 4. 10 The SNM in read state

If the SRAM cell is in the write state, the two sides of the circuit are different now. This time the access transistors M5 and M6 are enabled. The initial data stored in the SRAM cell is Q=0 (QB=1), now we want to write 1 into this memory cell. Thus, we set BL to logical 1 and BLB to logical 0. The circuit form is shown in Figure4.11. In this situation, we give Q and QB a static noise source, different characteristics could be presented. The side of the biggest square between those two characteristics is the SNM in this state.

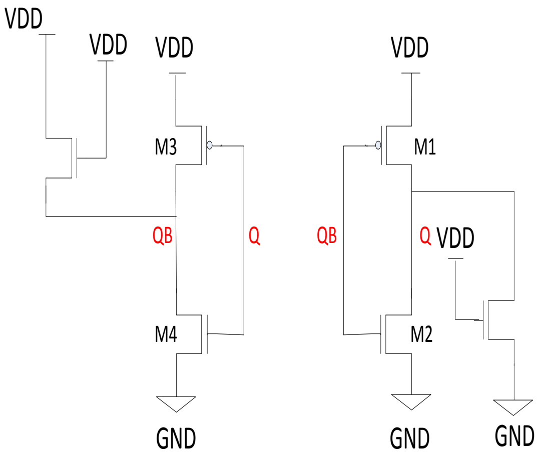


Fig 4. 12 The schematic of SNM simulation in write state

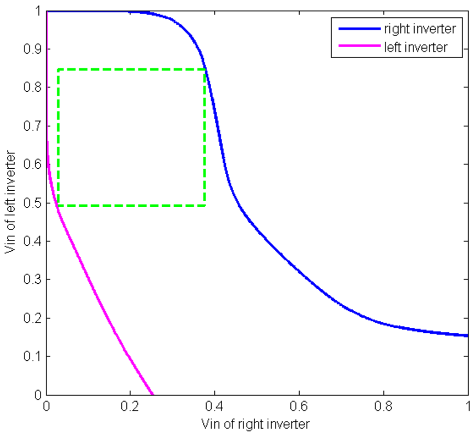


Fig 4. 11 The SNM of write state

5. Results

The results of the project got from the simulation will be presented in this section, and an analysis and discussion of the results related to the initial objectives will be explained.

In this project, two main parts of work were carried out. One is using the method of Statistical Blockade to do a very fast and efficient Monte Carlo simulation of rare circuit events on SRAM cell. The idea of further speeding up the simulation of the rare circuit events on SRAM by selecting the minimal classification threshold and minimal training samples is carried out. This is based on the the Statistical Blockade method. The second part of the work is to make a low power design of SRAM cell using multi-threshold voltage transistors. This method does not only reduce the leakage power of the SRAM cell, but also improves the Static Noise Margin of the memory cell under process variations.

How PVT variations affect the performance of the circuits have analyzed. The working principle of the standard Monte Carlo simulation has been reviewed. It is an accurate and basic technique for statistical analysis. The Statistical Blockade technique to speed up the simulation of rare circuit events was carried out. Moreover, the minimal classification threshold and the minimal number of training samples for the classifier were evaluated. Finally, the method of dual threshold voltage design was carried out to decrease the leakage power of the SRAM cell.

5.1 Analysis of PVT variation using the Standard Monte Carlo

As the influence of the variability of the parameters on the performance of the SRAM cell are desired to analyzed, the standard Monte Carlo method is carried out in this project. Different numbers of parameters that are varied with Gaussian distributions were applied in this part.

Initially, one parameter was varied in the simulation. Following the procedure of the standard Monte Carlo simulation, firstly, the testing circuit is an SRAM cell. Secondly, all parameter values are extracted randomly and use the respective Gaussian distributions. In this case, the parameter is the threshold voltage of the NMOS, and the temperature is held at 25℃ and the supply voltage is VDD=1.0V. In the next stage, a parameter can be applied to circuit to produce a particular instance. Then, these instances are simulated using HSPICE to measure the read time and write time of the SRAM cell. Finally, return to the second step and repeat it again and again for 1000 times. Then Matlab can be used to show all the results in a graph, so that we can easily see how the value of the read time or write time changes.

According to the High Performance nanometer model (provided on the Predictive Technology Model website), the nominal values of those parameters in the 45nm technology library of BPTM is L=45nm, W=90nm, the threshold voltage of NMOS (nvt)= 0.46893v, pvt= -0.49158v, and gate oxide thickness of NMOS (ntox)=1.2nm and ptox=1.3nm. The variations are considered as a Gaussian distribution that has a deviation of ±3σ around of their nominal values. The threshold voltage of NMOS is swept using the Monte Carlo simulation under the Gaussian distribution as follows:

nvt= agauss(0.46893v,0.093786v,3),

The Gaussian distribution of the input parameter which refers to the threshold voltage of the NMOS is shown in figure5.1 in this case. The proportion of variation is 20%, and the deviation is ±3σ. The initial value stored in the SRAM cell is Q=0 and QB=1.

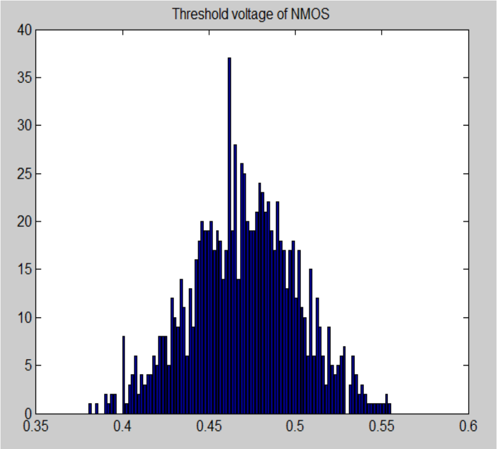


Fig 5. 1 The distribution of the threshold voltage of NMOS

After repeating simulation for 1000 times, the plotting results of the read time and write time by using Matlab are shown in figure 5.2. The read time and write time varies as Gaussian distributions.

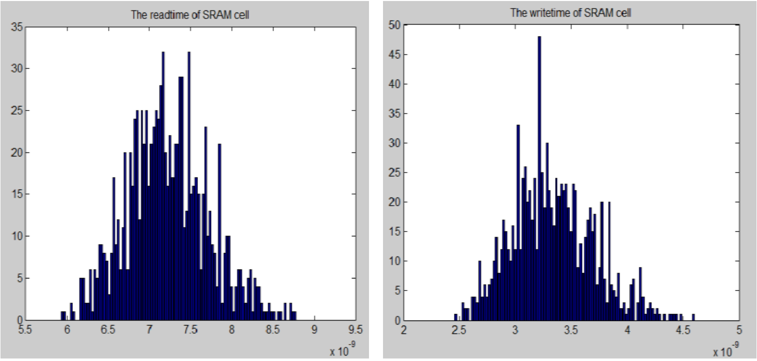


Fig 5. 2 The results of read time and write time with one parameter variation

Next, two varying parameters under Gaussian distributions were proposed (figure5.3), which refer to the threshold voltage of NMOS and PMOS. The Gaussian distributions of these two parameters are swept as follows:

nvt= agauss(0.46893v,0.093786v,3),
pvt= agauss(-0.49158v,0.098316v,3),

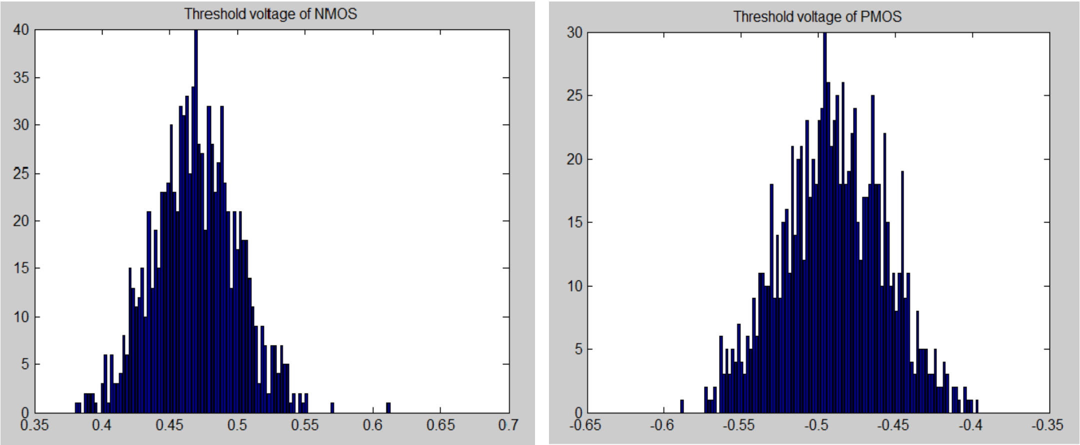


Fig 5. 3 The distributions for threshold voltage of NMOS and PMOS

The working conditions of the SRAM cell is the same as when we varied one parameter. The temperature is held at 25℃, and the supply voltage is VDD=1.0V. The initial value stored in the SRAM cell is also Q=0 and QB=1. The difference is two parameters were extracted this time. Then these instances were simulated using HSPICE to measure the read time and write time of the SRAM cell. After sweeping 1000 times, the variations of read time and write time got using Matlab is as follows:

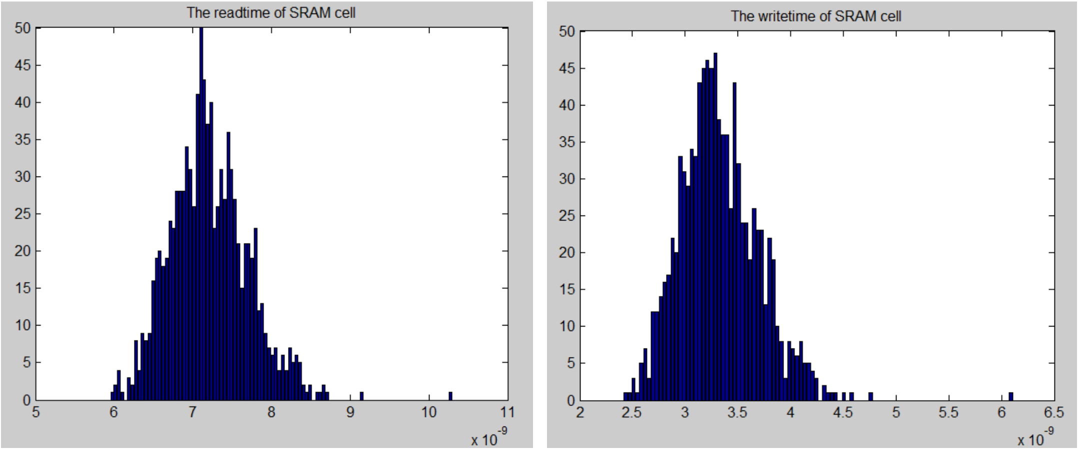


Fig 5. 4 The simulation results of read time and write time of the SRAM with two parameter variations

When three parameters are proposed, different distributions for read time and write time of the SRAM cell were presented. The Gaussian distributions are given as follows:

nvt= agauss(0.46893v,0.093786v,3),
pvt= agauss(-0.49158v,0.098316v,3),
ntox= agauss(1.2n,0.24n,3),

This time the three parameters we extracted are threshold voltage of NMOS, threshold voltage of PMOS, and the gate oxide thickness of NMOS. The deviation of the Gaussian distribution is ±3σ, and the variation is 20%. Figure5.5 shows the variations of these three parameters.

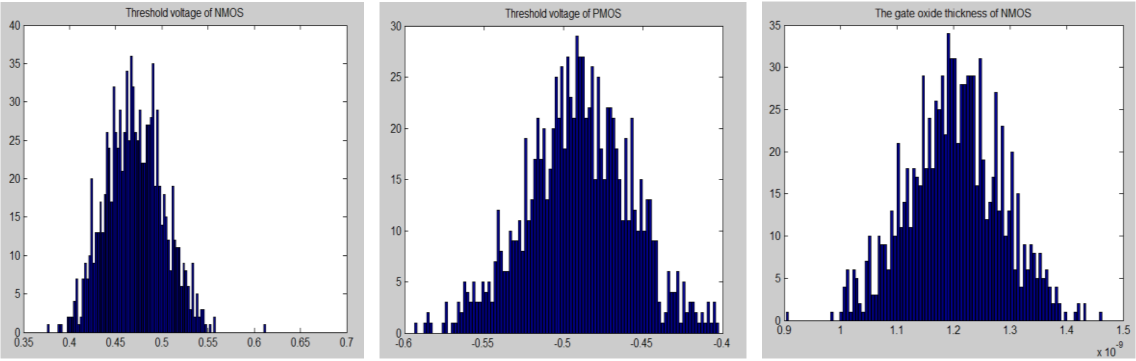


Fig 5. 5 The distributions of threshold voltage of NMOS, PMOS, and the gate oxide thickness of NMOS

The read time and write time of the SRAM cell are then determined by simulating those instances using HSPICE. After sweeping 1000 times, the variations of read time and write time got using Matlabe are as follows:

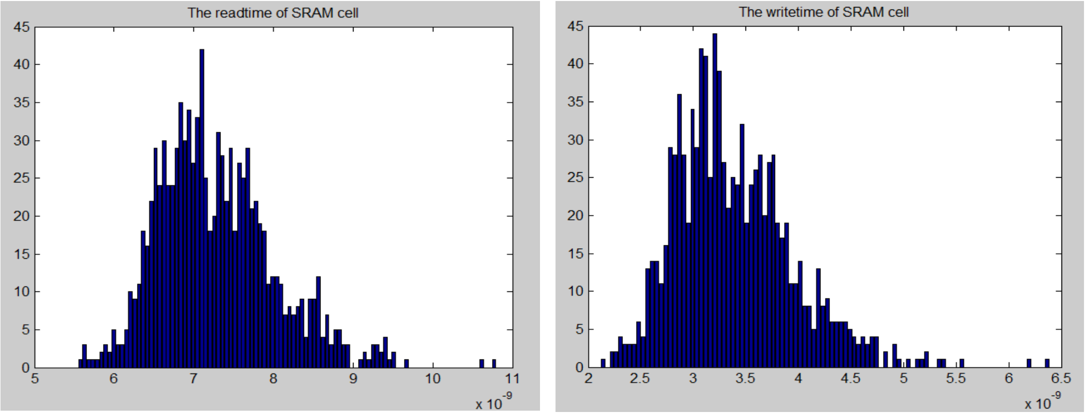


Fig 5. 6 The simulation results for the read time and write time of the SRAM cell under three parameter variations

Finally, four parameter variations were carried out to see if that can result in any difference on the read time and write time of the SRAM cell. The Gaussian distributions are given as follows:

nvt= agauss(0.46893v,0.093786v,3),
pvt= agauss(-0.49158v,0.098316v,3),
ntox= agauss(1.2n,0.24n,3),
ptox= agauss(1.3n,0.26n,3),

The four parameters we extracted this time are threshold voltage of NMOS, threshold voltage of PMOS, gate oxide thickness of NMOS, and the gate oxide thickness of PMOS. The Gaussian distribution of those four parameters got using Matlab are as follows:

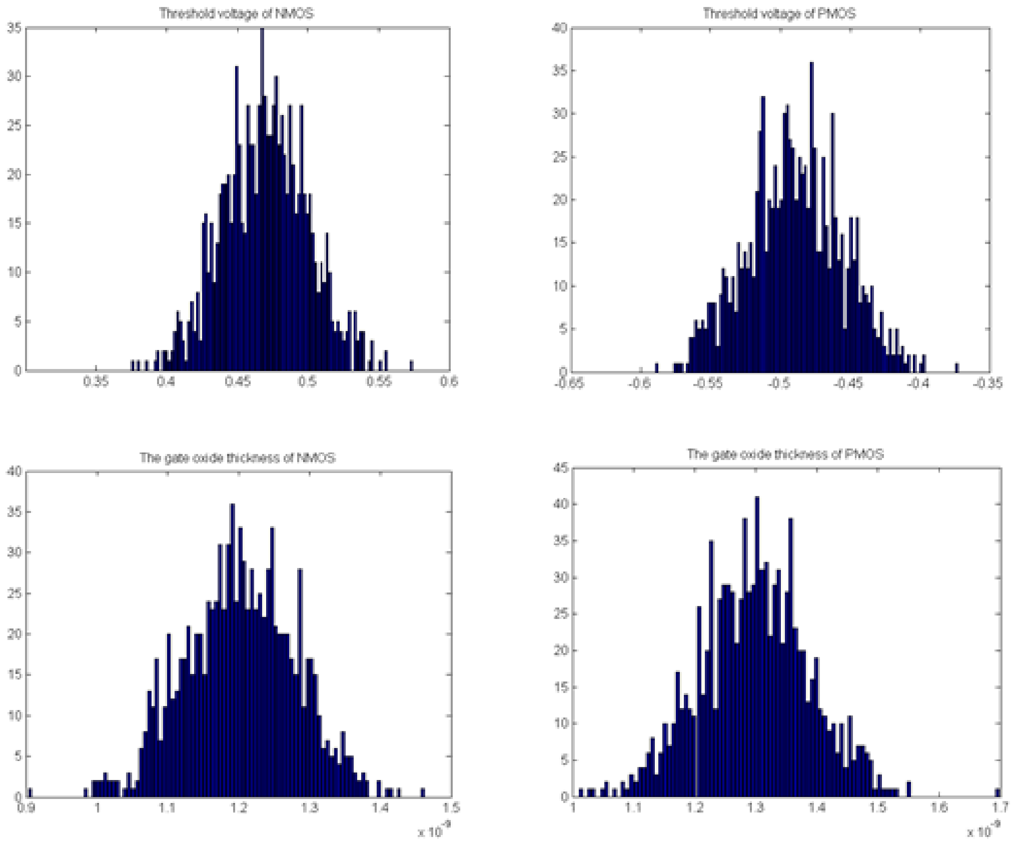


Fig 5. 7 The distributions of threshold voltage of NMOS, PMOS and the gate oxide thickness of NMOS, PMOS

After sweeping the parameters for 1000 times, the results of the simulations were given to Matlab. We got the distributions of the read time and write time of the SRAM cell as shown in Figure5.8.

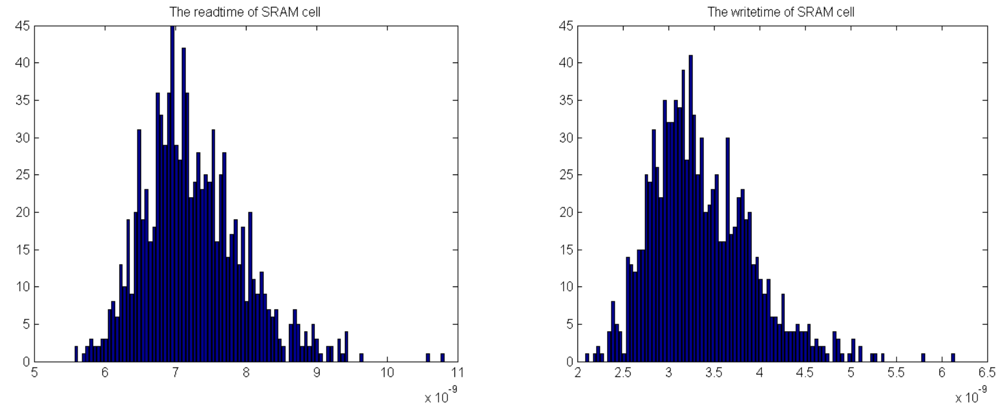


Fig 5. 8 The results of read time and write time of the SRAM cell under four parameter variations

Figure 5.8 was the plotting of statistical results. It indicated the read time and write time of the SRAM cell. The results can be used as the judgement for the performance of integrated circuit.

5.2 Results for Statistical Blockade

5.2.1 The idea of Statistical Blockade

The idea and the algorithm of the Statistical Blockade were expressed in section four. To implement the SB approach, 1000 samples were firstly simulated using HSPICE as the training samples of the classifier, and then the results of the output parameters and the tail threshold of the distribution were plotted(set it as 99%). This step is fast. Second, the training samples, the simulation results, and the classification threshold (we set it as 97%) were used to train and build the classifier. This step is also fast and returns the classifier. The step of training the classifier can be shown use figure5.9. Among the 1000 samples, the blue points are the tail points, the red points are the body points.

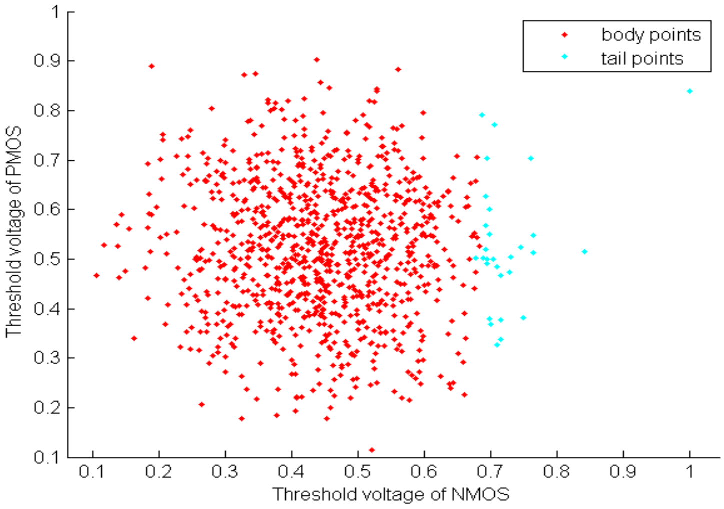


Fig 5. 9 Training the classifier

Third, 10,000 Monte Carlo samples were generated, these points are stored in the two-dimensional metrics as we consider the two parameter variations (the threshold voltage of PMOS and the threshold voltage of NMOS) with the Gaussian distributions. The deviation of the Gaussian distribution is ±3σ, and the variation is 20%. Then the sample points are classified using the classifier we have built, and the nontail points are blocked. From that, we have 349 tail points. Finally, these 349 tail points were simulated using HSPICE to detect the fail events. The classification of generated Monte Carlo samples is shown below in figure5.10.

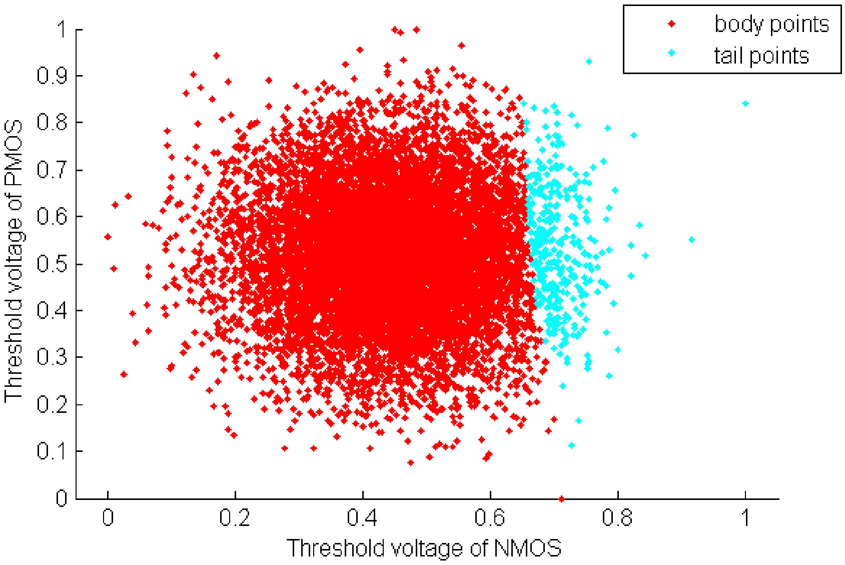


Fig 5. 10 The Classification of Generated Samples

5.2.2 Comparing the Statistical Blockade with the Standard Monte Carlo Simulation

The purpose of proposing the Statistical Blockade method is to make the simulation of statistical rare events more efficient when compared with the standard Monte Carlo simulation. Suppose that we generate 10000 samples, and want to find the points that would probably fail. Both the empirical method and Statistical Blockade are carried out to complete the work, and we will compare them.

In the method of Statistical Blockade, 1000 samples were used to train the classifier. The tail threshold and classification threshold are set to 99% and 97% respectively. After simulating, the conditional CDF of the tail points with both the empirical method and Statistical Blockade are compared. The simulation using Statistical Blockade can detect almost all of the tail points. Figure5.11 illustrates that these two CDFs represent a good match.

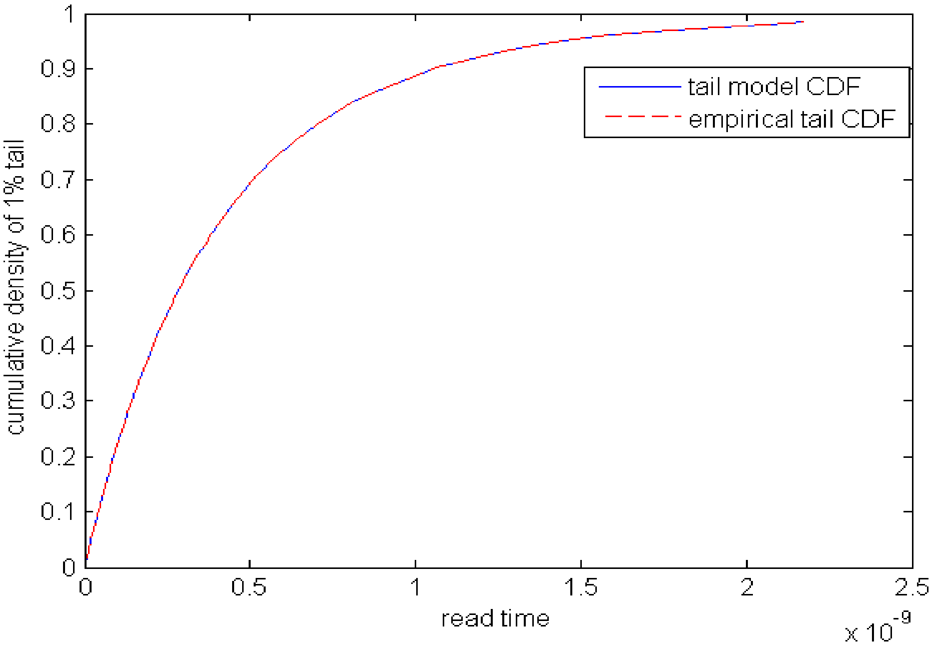


Fig 5. 11 Empirical vs. Statistical Blockade

However, the number of the simulations required to detect such tail points for these two methods are different. As illustrated in the table2, in the Statistical Blockade method, 349 tail points are picked up after classification. Hence, the total number of simulations used in Statistical Blockade is 1349 (1000+349). However, to detect the tail points of the samples, the standard Monte Carlo needs 10000 Monte Carlo simulations. And the method of Standard Blockade can cover all of the tail points.

Method	Standard Monte Carlo	Statistical Blockade
Number of simulation	10000	1349
Tail points covered	99	99

Table 2 The simulations times of Standard MC and SB and the tail points they covered

As a result, the method of Statistical Blockade not only can detect the tail points accurately, but also speed up the simulation of rare circuit events by reducing the number of simulations.

5.3 Evaluate the minimal classification threshold and training samples

To accomplish the idea of further speeding up the statistical rare events simulation, the total number of simulations is required to be reduced. Therefore, the minimal classification threshold and training samples that can ensure all of the tail points are covered were desired to be evaluated.

The 10000 samples were supposed to be generated, and the failure points among them respective to different tail region (1%, 2%, and 3%) were required to be detected. This means the tail thresholds (t) are set to 99%, 98%, and 97% respectively. The minimal classification threshold of the classifier is chosen using the second algorithm. The classification threshold varied from 99% to 97%, and it is increased by 1% for each time. The third algorithm is used to evaluate the minimal number of training samples. Hence, the number of training samples of the classifier was changed from 100 to 2000, and each time changed by 100 samples. Suppose that four-dimensional data was used to train the classifier. This means there are four parameters variations that follow the Gaussian distributions. Such Gaussian distributions have a deviation of ±3σ around their nominal values, and the variation is 20%. Table5.1 shows the minimal classification threshold and minimal training samples for each tail threshold. for an example, take the tail threshold of 99% after the classification threshold and training samples were minimized, the total number of simulation is only 1037 (800+237), where 237 is the tail points picked up after classification.

Tail threshold (t)	99%	98%	97%
Minimal classification threshold	98%	97%	96%
Minimal training samples	800	400	800

Table 3 The minimal classification threshold and training samples for different tail threshold

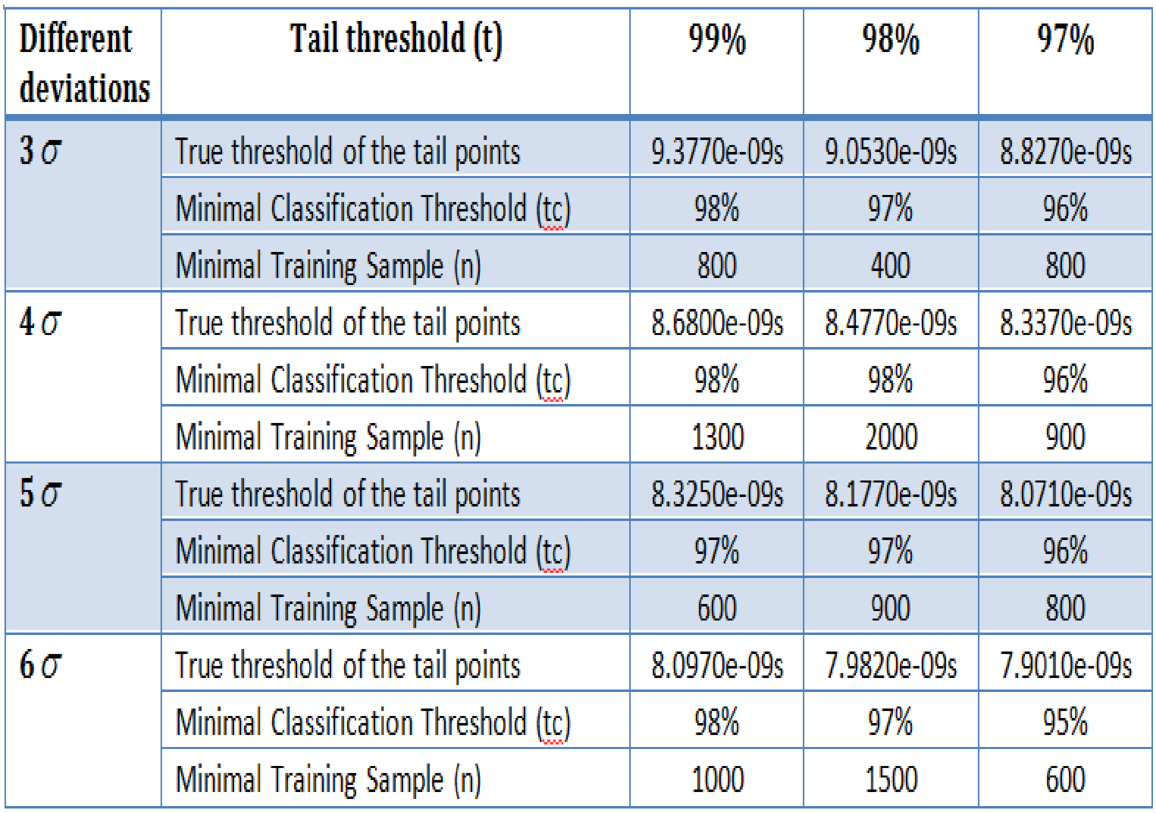
As the simulation is a cost of time, therefore, choosing the minimal classification threshold and training samples is a more efficient way to simulate the rare circuit events for SRAM cells.

5.4 Different deviations of parameters with Statistical Blockade

The idea of minimizing the classification threshold and the training samples of the classifier was accomplished to further speed up the simulation on rare circuit events. Wondering that if the deviation of each parameter was changed from 3σ to 6σ, and the Statistical Blockade method was used to sample and classify the tail points of each of them, is there any difference when the minimal classification threshold and minimal training samples required for each of them are investigated?

This part is an implementation of minimizing the classification threshold and training samples. The deviation of each parameter variation was changed from 3σ to 6σ, and simulated 10,000 samples using HSPICE. Next the idea of minimizing the classification threshold and training samples of the classifier was called upon respect to different tail thresholds (1%, 2%, 3%), and use this idea to choose the minimal classification threshold and training samples for each deviation. The results are shown in table4.

As we can see from table4, for the same deviation of the parameter, the minimal classification threshold or minimal training sample will increase when the tail threshold increases. And with the deviation of parameter increasing for the same tail threshold, the true threshold voltage of the tail points decreased.



Different deviations	Tail threshold (t)	99%	98%	97%
3 σ	True threshold of the tail points	9.3770e-09s	9.0530e-09s	8.8270e-09s
	Minimal Classification Threshold (tc)	98%	97%	96%
	Minimal Training Sample (n)	800	400	800
4 σ	True threshold of the tail points	8.6800e-09s	8.4770e-09s	8.3370e-09s
	Minimal Classification Threshold (tc)	98%	98%	96%
	Minimal Training Sample (n)	1300	2000	900
5 σ	True threshold of the tail points	8.3250e-09s	8.1770e-09s	8.0710e-09s
	Minimal Classification Threshold (tc)	97%	97%	96%
	Minimal Training Sample (n)	600	900	800
6 σ	True threshold of the tail points	8.0970e-09s	7.9820e-09s	7.9010e-09s
	Minimal Classification Threshold (tc)	98%	97%	95%
	Minimal Training Sample (n)	1000	1500	600

Table 4 True threshold of the tail points, minimal classification threshold, and minimal training sample required for different deviations of parameter

5.5 Low leakage SRAM design with dual threshold voltage transistors

As mentioned previously, three configurations of the dual threshold voltage design were proposed. C1, C2 and C3 refer to all of the transistors using high threshold voltage, four transistors using high threshold voltage, and two transistors using high threshold voltage respectively. The leakage current was measured use HSPICE for each configuration in every state. For example, the leakage current is measured for those three configurations in standby state with the increased threshold voltage. The trend of the reduction on the leakage current is shown in figure5.12.

As the leakage current is proportional to the leakage power, the trend of the leakage power is the same as the leakage current. Thus, the changes in the leakage current were measured. The leakage current reduction of each configuration was calculated as the threshold voltage increased. Refer to figure5.12, the leakage current reduction decreases as we move from C1 to C3. It also illustrate that the leakage current is decreasing for each configuration as the threshold voltage increased. Also, if the threshold voltage of all transistors within a cell is increased, the leakage current reduction is the highest.

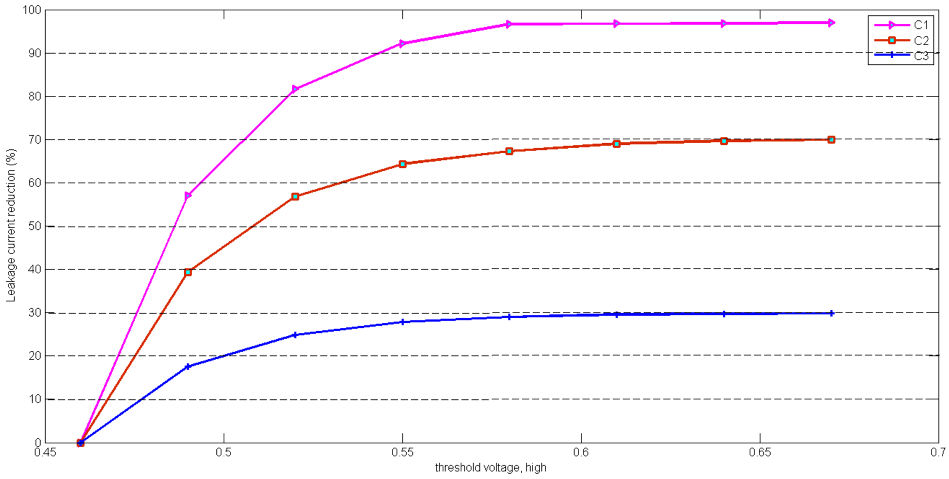


Fig 5. 12 The reduction of leakage current for each configuration in standby state

The reduction of the leakage current for the read operation and the write operation is measured as well. Figure 5.13 shows the changes in the leakage current reduction. As with the standby state, the leakage reduction is increased as the threshold voltage increases for each configuration. The leakage current reduction of the configuration C1 is the highest.

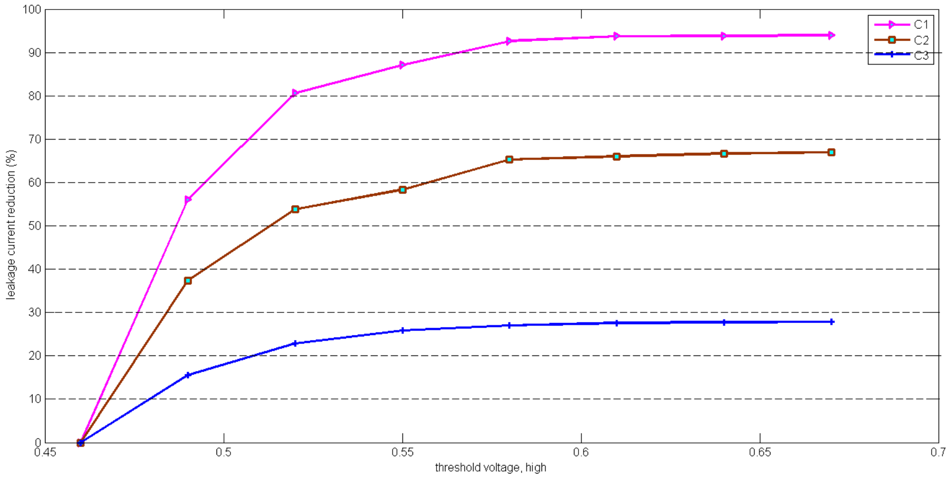


Fig 5. 13 The reduction of leakage current for each configuration in read operation

Figure 5.14 shows the changes in the leakage current reduction when an SRAM cell performs a write operation. It is the same as in the standby and read states: with the increase of the threshold voltage, the leakage current reduction increases for each configuration. The configuration C1 continues to have the highest leakage current reduction.

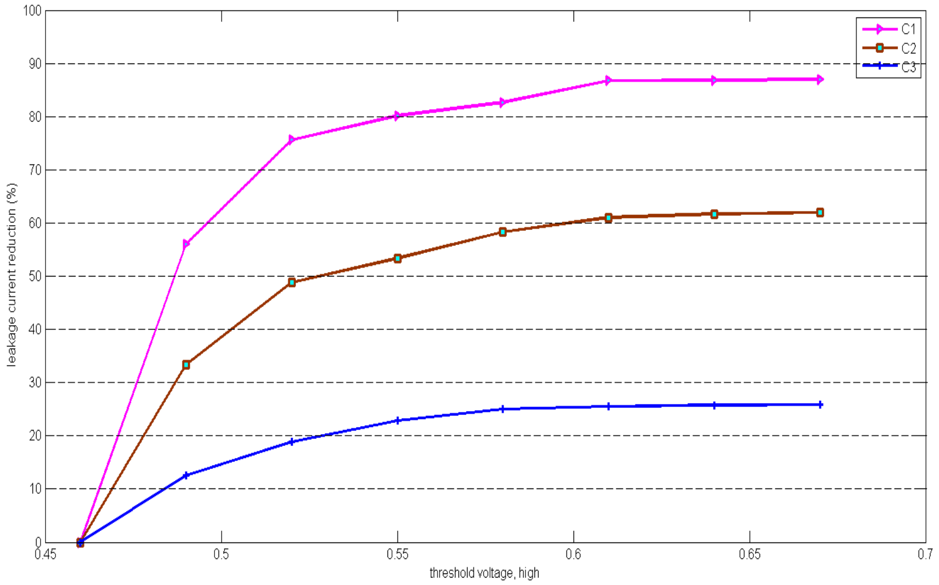


Fig 5. 14 The reduction of leakage current for each configuration in read operation

Figure5.12 to figure5.14 show that for all states, the leakage current reduction is increased with the increase of threshold voltage on high V_t transistors in each configuration. It provides a good result for a low leakage design.

However, the other output parameters, such as read delay, write delay, and Static Noise Margin of the SRAM cell are also very important factors of the circuit performance. Thus, a question emerges of how the low leakage design affects the read time, write time, and the SNM of the SRAM cell.

Then the variety of the read time and write time of each configuration was measured. The purpose of this is to investigate whether increasing the threshold voltage will influence the read time and write time of the SRAM cell. Here, take the standby state as an example. Figure5.15 shows that the increase in read time of an SRAM cell for configuration C1 and C2 is very high when the threshold voltage increases. However, there is a very small increase for the read time of configuration C3. This means the increase in threshold voltage will result in a noticeable influence in configurations C1 and C2, but will not have much effect on the read time of configuration C3.

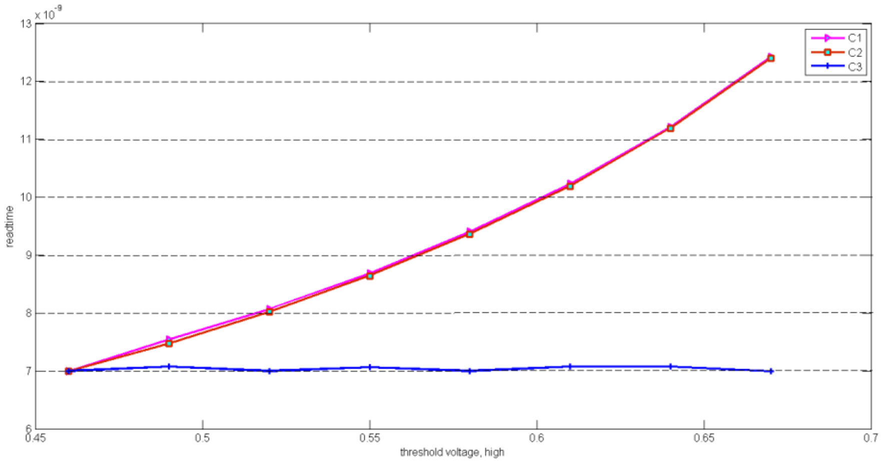


Fig 5. 15 The read time for each configuration in standby operation

Figure5.16 shows the changes of the write time for each configuration in the hold state. It illustrates that the write time of configuration C1 increased dramatically when the threshold voltage increased. However, not all of the configurations increase the write time, for example, the write time of the configuration C3 decreases slightly with the threshold voltage increased for the high V_t transistors.

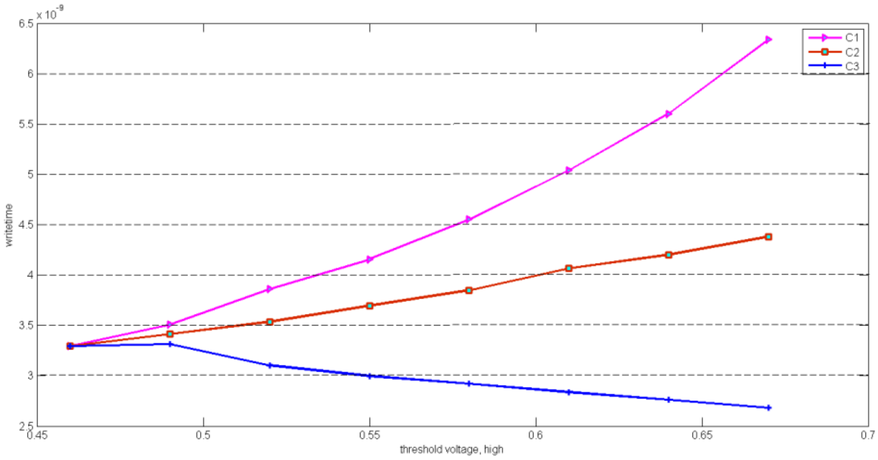


Fig 5. 16 The write time for each configuration in standby operation

As the Static Noise Margin is one of the most important factors of the performance of the SRAM cell, we also estimated the SNM for each of the configuration under process variation. We considered four varying parameters with the Gaussian distributions. The standard deviation of the Gaussian distributions is 3σ, and the variation is 20%. As seen in Figure5.16, the SNMs of configuration C1 and C2 are improved. However, the SNM of configuration C3 is slightly decreased.

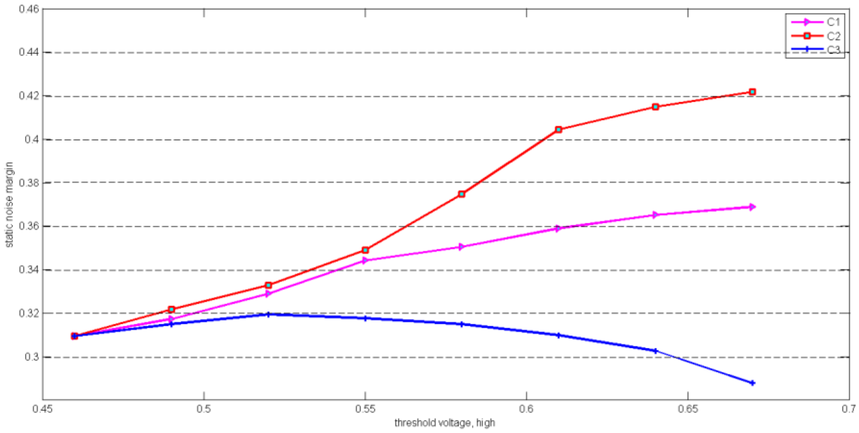


Fig 5. 17 The static noise margin for each configuration in standby operation

Compare the Static Noise Margin of those three configurations. For each configuration, the threshold voltage of the high threshold voltage transistors is 0.67V, the threshold voltage of the low threshold voltage transistors is 0.46V. In this case, we take the SNM in standby state as an exmple. After simulating, the graph of Static Noise Margin for each configuration can be shown in figure18. As we can see, configuration C1 provide the best Static Noise Margin. The SNM of configuration C3 is the smallest one.

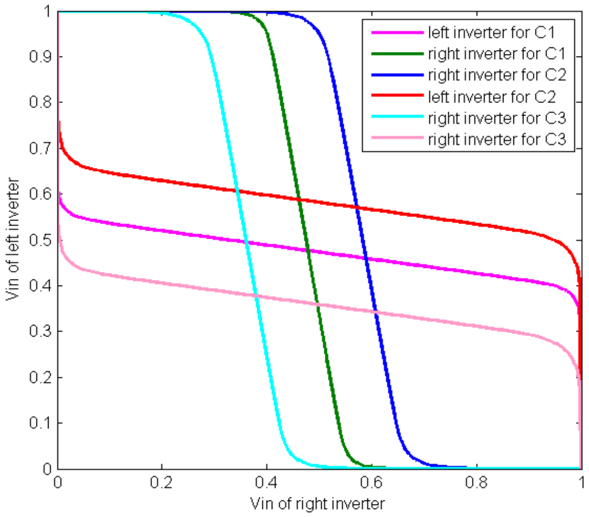


Fig 5. 18 The SNM of each configuration in standby state

To conclude, the increase of the threshold voltage for high V_t transistors in each configuration will result in the increase of the leakage reduction. The Static Noise Margin is also improved dramatically with configuration C1 and C2. However, the read time and write time of configuration C1 and C2 go up with the increasing threshold voltage.

6. Critical Evaluation

The critical evaluation of the work have done refer to each objectives is explained in this section.

- Analysis of PVT variation on SRAM cell using Statistical Blockade method, and compare this approach with standard MC.

The first objective of the project is implementation of a method that could do an efficient simulation of rare circuit event for SRAM cells. This was completed by analyzing the PVT variation on SRAM cells using the Statistical Blockade method. We successfully built a classifier using SVM to classify the large amount of samples for such a high-replication circuit like SRAM, and blocked the non-tail points among them. Therefore, only the tail points filtered by the classifier required to be simulated. In this way, the number of simulations is reduced, and therefore an optimized simulation of rare events was developed. Comparing the Statistical Blockade with the standard Monte Carlo simulation, it presents a good match of detected tail points, and more importantly, it provides us an efficient way of simulating the critical events of SRAM cells. The technique of increasing the weight of the tail points is carried out to make the classifier bias to the tail part of the distribution, this will help us detect more tail points to improve the accuracy.

- Propose a method of optimization of SB using minimal classification threshold and training samples of the classifier.

Then idea of further speeding up the analysis of the rare circuit events was proposed. This is achieved by evaluating the minimal classification threshold and training samples, which still ensures all of the tail points are covered. As the classification threshold and the training samples can decide how many samples are required to be simulated using HSPICE, the minimal value of these two parameters could further reduce the number of simulation. This means the simulation time could be further optimized to decrease the cost of the circuits.

- Change the deviation of the Gaussian distribution to investigate the variation of the minimal classification threshold and the training sample.

The deviation of the Gaussian distribution for each parameter was widened in the third stage of the project to investigate that what will happen on the value of minimal classification threshold and training samples of the classifier. As the decrease of the transistor scale may cause an increase of the process variation, this phenomenon can be considered as an increase in the deviation of the Gaussian distribution. Thus, analysis of the influence of the widened deviation is required. The deviation of the parameters was firstly increased. Then the Statistical Blockade method and the idea of evaluating the minimal classification threshold and training samples are proposed. The minimal classification and training samples are estimated with respect to each parameter deviation. As a result, for the same deviation of parameter, the minimal classification threshold or minimal training sample will increase with the increased tail threshold. And if the deviation of the parameter increases, for the same tail threshold, the true threshold voltage of the tail points decreases.

All of the work mentioned above is based on the method of Statistical Blockade. In this part, we have achieved all of the objectives, and addressed the questions we proposed. In practice, there are more questions we need to consider. In this project, for example, as the influence between the parameters is not noticeable, all of the parameter variations are considered to be independent to each other. If the parameters are not independent to each other, how will this influence the results we simulated?

- Propose a low leakage design on SRAM cell with the dual threshold voltage.

The final objective is to make a low leakage design on SRAM cells. This is accomplished by using the idea of dual threshold voltage transistors. Increasing the threshold voltage of some transistors could reduce the current through them. As analyzed previously, the leakage power for all working states of SRAM cell was reduced by the design with dual threshold voltage transistors. As the read delay, write delay, and the Static Noise Margin are also the most important factors of an SRAM cell, we also measured all of them in respect to the design with dual threshold voltage transistors. It can be found that not all configurations would influence the read time and write time by increasing of the threshold voltage. And the results provide an improvement of the SNM of the memory cell.

As the dynamic power is still an important part of the total power dissipation, the analysis of dynamic power is required. In the future, the evaluation of dynamic power with the dual threshold voltage transistor will be carried out in order to investigate the effect of the dual threshold voltage design on dynamic power. By analyzing both the dynamic power and static power, a better assignment of the transistor and a better value of threshold voltage can be selected to make a design that can both reduce the leakage power and dynamic power.

7. Future work

7.1 Statistical Blockade method

In the analysis of the PVT variation on SRAM cells using Statistical Blockade, all the parameters are considered as independent with each other. This means if one parameter various, it would not influence the others. It is an ideal circumstance. Hence, there are two questions come out: Is there any relationship between the different parameter variations? And if those parameter variations are not independent, how this situation affects the analysis using Statistical Blockade? For example, if the threshold voltage of the NMOS is various follow a Gaussian distribution, there might be an influence on the threshold voltage of the PMOS.

As we analyzed above, the relationship of all parameters are need to be concerned. Firstly, for each time, one parameter variation could be proposed, other parameters would kept with normal values. Secondly, two parameters could be varied with Gaussian distributions. Then, any difference of the results can be detected as we measured with one parameter changes and two parameter changes. If there is any difference, different parameters would affect each other. The measurement of how they affect each other also can be evaluated.

7.2 Low-leakage design of SRAM cell with Dual V_t transistors

In this project, three kinds of symmetric configurations of the high threshold voltage assignment were proposed. For all of the configurations, assume that the threshold voltage of each transistor can be considered as independent of each other, even if the channel doping is changed. This assumption seems to be safe, because the channels of the transistors are not too close to each other in the SRAM cell.

However, it can be shown that if only one threshold voltage is used in the SRAM cell, the leakage power of the circuit also can be reduced. Hence, there should be some effects between the threshold voltages of the transistors which should be measured as well.

The dynamic power is also one of the important parts of power consumption for one circuit. Therefore, how the dual threshold voltage design influences the dynamic power should be taken into consideration. The dynamic power for under each configuration should be measured as well. Then we can consider most of the factors of SRAM cell, which refers to read time, write time, Static Noise Margin, and both Dynamic Power and leakage power. In this way, a better design will be presented which has smaller dynamic power and leakage power consumption, larger Static Noise Margin, and also appropriate read time and write time at the same time.

8. Conclusion

We have observed that circuits like SRAM are highly replicated, there may be millions of cells in one chip. Therefore, rare events of the cell will induce a not-so-rare event of the whole system. To detect the rare events of the cell efficiently is of growing concern. As the Standard Monte Carlo is too slow to simulate such a large amount of samples, a new problem of how to efficiently simulate the rare circuit events in such a highly replicated circuit is presented.

It is recognized that the method of Statistical Blockade could solve this problem. The Statistical Blockade approach can do a very fast simulation of the critical circuit event. As the simulation using HSPICE is costly, the number of simulations needs to be reduced. This is achieved by efficiently filtering the tail points of the distribution using the classifier we built. We do not need to simulate all of the samples, only the tail points which are supposed to be the failure events are required to be simulated.

The idea of minimizing the classification threshold and training samples was proposed to further speed up the simulation of the rare circuit events. Based on the idea of Statistical Blockade, all of the samples we need to simulate are the training samples of the classifier and the tail points we filtered. If we could reduce the number of those samples, the statistical analysis of the rare events would be faster. This was achieved by evaluating the minimal classification threshold and the training samples of the classifier.

As we always want a low power design, we need to find a way to reduce the power consumption. Recently, leakage power is playing an important role in the total power dissipation. The design with dual threshold voltage transistors is proposed to reduce leakage power. And this design can improve the Static Noise Margin. The read time and write time of the memory cell can be controlled to not be overhead.

As a conclusion, the Statistical Blockade is a basic approach to do an efficient simulation of rare circuit events. Based on this method, minimizing the classification threshold and training samples can be proposed to further speed up the statistical simulation of critical events. The dual threshold voltage design is an efficient approach to reduce leakage power.

9. Bibliography

[1] Iizuka, T.; , "Embedded memory: a key to high performance system VLSIs," VLSI Circuits, 1990. Digest of Technical Papers., 1990 Symposium on , vol., no., pp.1-4, 7-9 Jun 1990.

[2] S. P. Mohanty, E. Kougianos, and D. K. Pradhan, "Simultaneous scheduling and binding for low gate leakage nano-complementary metal-oxide-semiconductor data path circuit behavioural synthesis", IET Computers & Digital Techniques (CDT), Volume 2, Issue 2, pp. 118-131, 2008.

[3] Singhee, A.; Rutenbar, R.A.; , "Statistical Blockade: A Novel Method for Very Fast Monte Carlo Simulation of Rare Circuit Events, and its Application," Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE '07 , vol., no., pp.1-6, 16-20 April 2007.

[4] Sohaib Majzoub, Resve Saleh, and Rabab Ward, “PVT Variation Impact on Voltage Island Formation in MPSoC Design”, 10th Int'l Symposium on Quality Electronic Design, 2009 IEEE.

[5] Kyung Ki Kim; Yong-Bin Kim; , "A Novel Adaptive Design Methodology for Minimum Leakage Power Considering PVT Variations on Nanoscale VLSI Systems," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.17, no.4, pp.517-528, April 2009.

[6] Nassif, S.R.; , "Modeling and analysis of manufacturing variations," Custom Integrated Circuits, 2001, IEEE Conference on. , vol., no., pp.223-228, 2001

[7] Jan M. Rabaey, “Advanced Digital Integrated Circuits lecture”, 203 McLaughlin, 2006.

[8] Duane Boning and Sani Nassif, “Models of Process Variations in device and interconnect”, pp 98-116, Design of High-Performance up Circuits, 2000.

[9] Kanak Agarwal and Sani Nassif, “Statistical Analysis of SRAM Cell Stability”, IBM research, 2006.

[10] Sandeep, R.; Deshpande, N.T.; Aswatha, A.R.; , "Design and Analysis of a New Loadless 4T SRAM Cell in Deep Submicron CMOS Technologies," Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on , vol., no., pp.155-161, 16-18 Dec. 2009.

[11] Onaissi, S.; Najm, F.N.; , "A Linear-Time Approach for Static Timing Analysis Covering All Process Corners," Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on , vol., no., pp.217-224, 5-9 Nov. 2006.

[12] Bufler, F.M.; Asahi, Y.; Yoshimura, H.; Zechner, C.; Schenk, A.; Fichtner, W.; , "Monte Carlo simulation and measurement of nanoscale n-MOSFETs," Electron Devices, IEEE Transactions on , vol.50, no.2, pp. 418- 424, Feb. 2003.

[13] Kerwin Khu, “Statistical Modeling for Monte Carlo Simulation using Hspice”, SNUG Singapore, 2006.

[14] F. M. Bufler, Yoshinori Asahi, Hisao Yoshimura, Christoph Zechner, A. Schenk, and

Wolfgang Fichtner, “Monte Carlo Simulation and Measurement of Nanoscale n-MOSFETs”, 2003 IEEE.

[15] Okyere Attia, J.; , "Teaching AC circuit analysis with MATLAB," Frontiers in Education Conference, 1995. Proceedings., 1995 , vol.1, no., pp.2c6.9-2c612 vol.1, 1-4 Nov 1995.

[16] Singhee, A.; Jiajing Wang; Calhoun, B.H.; Rutenbar, R.A.; , "Recursive Statistical Blockade: An Enhanced Technique for Rare Event Simulation with Application to SRAM Circuit Design," VLSI Design, 2008. VLSID 2008. 21st International Conference on , vol., no., pp.131-136, 4-8 Jan. 2008.

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

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

[19] Y. Kilic, and M. Zwolinski, “Behavioral fault modeling and simulation using VHDL-AMS to speed-up analog fault simulation,” Analog integrated circuits and signal processing, vol. 39, pp. 177–190, May, 2004.

[20] Mahesh Mamidipaka, Kamal Khouri, Nikil Dutt, and Magdy Abadir, “Leakage Power Estimation in SRAMs”, CECS Technical Report ,pp.03-32, 2003.

[21] N. Azizi et al, “Low-leakage asymmetric-cell SRAM,” IEEE Trans. on VLSI Systems, vol. 11, no. 4, Aug. 2003, pp.701-715.

[22] C. H. Kim et al, “A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations,” IEEE Trans. on VLSI Systems, vol. 13, no.3, Mar. 2005, pp. 349-357.

[23] C. Kim and K. Roy, “Dynamic Vt SRAM: a leakage tolerant cache memory for low voltage microprocessor,” in Proc. ISLPED, 2002, pp. 251–254.

[24] Evert Seevinck, Frans J. List, and Jan Lohstroh, “Static-Noise Margin Analysis of MOS SRAM Cells”, IEEE Journal of Solid-State Circuits, VOL. SC-22, NO. 5, October 1987.

[25] Makosiej, A.; Vladimirescu, A.; Thomas, O.; Amara, A.; , "An SNM estimation and optimization model for ULP sub-45nm CMOS SRAM in the presence of variability," NEWCAS Conference (NEWCAS), 2010 8th IEEE International , vol., no., pp.337-340, 20-23 June 2010.

[26] List, F. J., "The Static Noise Margin of SRAM cells," Solid-State Circuits Conference, 1986. ESSCIRC '86. Twelfth European , vol., no., pp.16-18, 16-18 Sept. 1986.

[27] T. Joachims, Making Large-Scale SVM Learning Practical. Universit¨at Dortmund: LS8-Report, 24, 1998.

[28] C. C. Chang and C. J. Lin, LIBSVM: a Library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm/, 2001.

[29] T. Joachims, “Making large-scale svm learning practical,” MIT Press, 1998.

[30] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and

Techniques. San Francisco: Morgan Kaufmann, 2005.

[31] Luo Sun; Mathew, J.; Dhiraj, K.P.; Saraju, P.M.; , "Algorithms for rare event analysis in nano-CMOS circuits using statistical blockade," SoC Design Conference (ISOCC), 2010 International , vol., no., pp.162-165, 22-23 Nov. 2010

[32] Doorn, T.S., ter Maten, E.J.W., Croon, J.A.; Di Bucchianico, A., Wittich, O., , "Importance sampling Monte Carlo simulations for accurate estimation of SRAM yield," Solid-State Circuits Conference, 2008. ESSCIRC 2008. 34th European , vol., no., pp.230-233, 15-19 Sept. 2008

Appendix: Source code

The operation of single 6T SRAM cell (source code in HSPICE)

```
***Definition of parameters
.param cload=5pf
.param vdd=1
.param l=45n
.param w=90n
 VCC 1 0 'vdd'
.IC V(Q)=0  ***The initial data stored in SRAM cell
.IC V(QB)=1

.param nvt= agauss(0.46893v,0.093786v,3) ***The Gaussian distribution of threshold voltage
+pvt= agauss(-0.49158v,0.098316v,3)
.param ntox=1.2n
.param ptox=1.3n

.GLOBAL 1
***Input data pulse
VD  D  0 pulse 0 'vdd' 100n 1n 1n 150n  400n

***Power supply of the circuit
VPC PC 0 pulse 0 'vdd'  50n 1n 1n 40n  100n
VWL WL 0 pulse 0 'vdd' 110n 1n 1n 25n  100n
VWE WE 0 pulse 0 'vdd' 110n 1n 1n 25n  200n
VSAE SAE 0 pulse 0 'vdd' 235n 1n 1n 25n  200n

***Precharge circuit
XIPC PC PCB INV
M_p2  BL  PCB 1 1  PMOS W='w*15' L='l'
M_p1 BLB PCB 1 1  PMOS W='w*15' L='l'
***The connection of 6 transistors cell
M1 Q   QB  1    1 PMOS W='w'    L='l'
M2 Q   QB  0    0 NMOS W='w*1.5' L='l'
M3 QB  Q   1    1 PMOS W='w'    L='l'
M4 QB  Q   0    0 NMOS W='w*1.5' L='l'
M5 Q   WL  BL  0 NMOS W='w'    L='l'
M6 QB  WL  BLB 0 NMOS W='w'    L='l'
***Write driver of the circuit
M_n4  BL  WE  Q1  0  NMOS W='w*20' L='l'
M_n3 BLB WE  Q2  0  NMOS W='w*20' L='l'
XIOUT D DB INV
M_n1  Q1  DB  0  0  NMOS W='w*20' L='l'
M_n2 Q2  D   0  0  NMOS W='w*20' L='l'
***Sense amplifier
M_qp1 SOB SAE 1 1    PMOS W='w*20'    L='l'
M_qp2 SOB SO 1 1PMOS W='w*20'    L='l'
M_qp3 SO SOB 1 1PMOS W='w*10'    L='l'
M_qp4 SO SAE 1 1PMOS W='w*10'    L='l'
M_qn1 S1 SAE 0 0NMOS W='w*30'    L='l'
M_qn2 S1 BLB S2 0    NMOS W='w*30'    L='l'
M_qn3 S1 BL S3 0NMOS W='w*30'    L='l'
M_qn4 S2 SO SOB 0    NMOS W='w*30'    L='l'
M_qn5 S3 SOB SO 0    NMOS W='w*30'    L='l'
***Sub-circuit used in the design
.SUBCKT INV in out1
MI1 out1 in 1 1 PMOS W='w*2' L='l'
MI3 out1 in 0 0 NMOS W='w' L='l'
.ENDS INV
***Load capacitance in bit line and bit line bar
C1  BL  0 cload
C2 BLB 0 cload
***Transient analysis from 0.1ns to 1000ns with the sweep of 1000 times
.TRAN 0.1n 1000n SWEEP MONTE=1000
***The measurement of read time and write time of the SRAM cell
.MEASURE ReadTime    trig v(WL) VAL='0.5*vdd'  rise=2
+                  targ v(BLB)   VAL='0.9*vdd' fall=2
```

```
.MEASURE WriteTime   trig v(WL) VAL='0.5*vdd'  rise=1
+                  targ v(QB)   VAL='0.5*vdd'  fall=1
.option  nopage nomod  post
```

The implementation of the Statistical Blockade (source code in Matlab)

```
%This code present the development of the classifier in the method of the Statistical Blockade,
with the tail threshold of 1% and classification threshold of 97%

%Define the tail size, the classification threshold is 97%
s=size(readtime_i,1);
tail_size = ceil(s*0.03);
thres_hole = readtime_i(tail_size);

%Build the classifier
classifier = zeros(s,1);
for i=1:s
   if (readtime_i(i) > thres_hole)
      classifier(i)=1;   % readtime in tail part is 1
   else
      classifier(i)=0;  % block the body points use 0
   end
end

%Propose the weight scale to increase the weight of the tail points
weight_scale = zeros(s,1);
weight_tail_size = floor(1*tail_size);
for i=1:weight_tail_size
   weight_scale(i) = 2.5;
end
for i=weight_tail_size:s
   weight_scale(i) = 1;
end
%Train the classifier
svmStruct = svmtrain(weight_scale,classifier,data,'-c 2^15 -g 2^3 -t 2');
[predict_label, accuracy, dec_values] = svmpredict(classifier, data, svmStruct);
test_mod = svmStruct;
test_classifier = classifier;

pre_label = zeros(10000,1);
for i=1:100:10000
   pre_label(i) = 1;
end
[predict_label, accuracy, dec_values] = svmpredict(pre_label, data_v, test_mod);

fid1 = fopen('readtime_3d_10000.txt','w');
for i=1:10000
   if predict_label(i)==1
      ex_ntox(i)=raw_ntox(i);
      ex_ptox(i)=raw_ptox(i);
      ex_nvt(i)=raw_nvt(i);
      ex_pvt(i)=raw_pvt(i);

      fprintf(fid1,        '%12.15f        %12.15f        %12.15f        %12.15f\n',
raw_ntox(i),raw_ptox(i),raw_nvt(i),raw_pvt(i));
   else
   ex_ntox(i)=0;
   ex_ptox(i)=0;
   ex_nvt(i)=0;
   ex_pvt(i)=0;
   end
end
fclose(fid1);
```

The measurement of SNM for each state (source code in HSPICE)

```
********************************************************** in standby state
VWL WL 0 0
Vin Q 0 'vdd'
**6 transistors cell
M1 Q   QB  1    1 PMOS W='w'     L='l'
M2 Q   QB  0    0 NMOS W='w*1.5' L='l'
M3 QB  Q   1    1 PMOS W='w'     L='l'
M4 QB  Q   0    0 NMOS W='w*1.5' L='l'
M5 Q   WL  0  0 NMOS W='w'     L='l'
M6 QB  WL  0 0 NMOS W='w'     L='l'
**********************************************************in read state
VWL WL 0 'vdd'
Vin QB 0 'vdd'
**6 transistors cell
M1 Q   QB  1    1 PMOS W='w'     L='l'
M2 Q   QB  0    0 NMOS W='w*1.5' L='l'
M3 QB  Q   1    1 PMOS W='w'     L='l'
M4 QB  Q   0    0 NMOS W='w*1.5' L='l'
M5 Q   WL  1  0 NMOS W='w'     L='l'
M6 QB  WL  1 0 NMOS W='w'     L='l'
**********************************************************in write state
VWL WL 0 'vdd'
Vin QB 0 'vdd'
**6 transistors cell
 M1 Q   QB  1    1 PMOS W='w'     L='l'
 M2 Q   QB  0    0 NMOS W='w*1.5' L='l'
 M3 QB  Q   1    1 PMOS W='w'     L='l'
 M4 QB  Q   0    0 NMOS W='w*1.5' L='l'
 M5 Q   WL  1      0 NMOS W='w'     L='l'
 M6 QB  WL  0      0 NMOS W='w'     L='l'

******DC analysis of the points Q and QB
.DC Vin 0 vdd 0.001
.prob Iin=par('0-I(Vin)')
```