EXECUTIVE SUMMARY:

Due to the effect of process variation, SRAM cell reliability is one of the major concerns in SRAM cell design. Scaling of devices and reducing the supply voltage have been highly beneficial in terms of integration density, operating speed, and lower power dissipation. Now, variability is a major challenge. This thesis addresses the following issue: Designing an embedded Nano-CMOS SRAM system, which not only consumes less power, but is also reliable and secure. The study and evaluation is narrowed down into different design levels namely, device level, architecture level, circuit level, and algorithm level.

Device level study includes the implementation of the SRAM cell designs by using different CMOS device models available. This enables us to actualize the behavior of the electrical devices based on their fundamental properties such as doping profiles, oxide thickness etc.

Architecture level analysis involves a detailed study of the components that comprise an SRAM cell and the factors that influence system reliability. At this stage, different cell designs are proposed, (which have been adopted from the papers [13-16]) to improve performance and reliability.

Circuit level assessment involves evaluating the performance of the proposed memory cells, by validating those using extensive HSPICE and MATLAB simulations, by subjecting them under the effect of process variation.

Algorithm Level design incorporates the evaluation of different error correction techniques, and a thorough discussion on the advantages and drawbacks of each technique and the trade-offs at hand has been done.

- The research carried out and reported in this thesis consists of a comparison and evaluation of five SRAM cell designs (See pages 27-34) and their stability and performance have been analyzed using the static noise margin parameter.
- Monte Carlo simulations of the SRAM cells were performed under the effect of process variation to study their stability and consistency in terms of read time, write time and power consumed.
- Multiple bit error correction using Hamming Code has been implemented, along with the evaluation of other error correction coding techniques available.

TABLE OF CONTENTS

EXECUTIVE SUMMARY……………………………………………………………………………1

DECLARATION…………………………………………………………………………………2

ACKNOWLEDGEMENTS……………………………………………………………………..3

LIST OF FIGURES…………………………………………………………………………...7

LIST OF TABLES……………………………………………………………………………….9

CHAPTER 1: INTRODUCTION……………………………………………………………..10

1.1 Aims………………………………………………………………………………10

1.2 Objectives …………………………………………………………………………....10

1.3 Background and context…………………………………………………………10

1.4 Thesis Outline……………………………………………………………………..11

CHAPTER 2: SRAM ARCHITECTURE……………………………………………….…12

2.1 About SRAMs……………………………………………………………………..12

2.2 Types of SRAM…………………………………………………………………...12

2.3 Power Consumption……………………………………………………………....12

2.4 Reliability ……………………………………………………………………….....12

2.5 Security…………………………………………….…………………………...13

2.6 Process Variations………………………………………………………………....13

2.7 Support Circuitry …………………………………………………………….....14

2.8 Embedded SRAM memory cells……………………………………………….…15

CHAPTER 3: DEVICE MODELS……………………………………………………….…15

3.1 Semiconductor device modeling……………………………………………………..15

3.2 SRAM scaling……………………………………………………………………..16

CHAPTER 4: STANDARD SIX TRANSISTOR (6T) SRAM…………………………………...18

4.1 6T SRAM cell architecture…………………………………………………………18

4.2 Standby mode……………………………………………………………….……..…18

4.3 Reading mode ………………………………………………………………………..19

4.4 Writing mode ………………………………………………………………………..19

4.5 Layout of a standard 6T SRAM cell……………………………………………....20

4.6 Peripheral Circuitry…………………………………………………………………..21

4.7 Sense Amplifier and bit-line pre-charge………………………………………………..21

4.8 Word Pre-charge..………………………………………………………………………22

4.9 Write Driver…………………………………………………………………………….23

CHAPTER 5: SIZING REQUIREMENTS IN SRAM CELL……………………… ………...24

5.1 Sizing and its effects………………………………………………………………....24

5.2 Trade-offs in SRAM design……………………………………………………………25

5.2.1 Area vs. yield ………………………………………………………………....25

5.2.2 Hold margin …………………………………………………………………..25

5.2.3 Read margin………………………………………………………………...26

5.2.4 Write margin ……………………………………………………………………26

5.2.5 Access time ………………………………………………………………...26

5.3 SRAM bit cell topologies ………………………………………………………...26

CHAPTER 6: SRAM CELL IMPLEMENTATION…………………………………………27

6.1 7T SRAM Memory Cell ………………………………………………………….27

6.1.1 Write …………………………………………………………………………….27

6.1.2 Hold ……………………………………………………………………………..27

6.1.3 Read ……………………………………………………………………………..28

6.2 8T SRAM Memory Cell…………………………………………………………..29

6.2.1 Write …………………………………………………………………………….29

6.2.2 Hold ……………………………………………………………………………..30

6.2.3 Read ……………………………………………………………………………..30

6.3 9T SRAM Memory Cell………………………………………………………..31

6.3.1 Write ……………………………………………………………………………31

6.3.2 Hold ……………………………………………………………………………..31

6.3.3 Read...31

6.4 10T SRAM Memory Cell …………………………………………………....33

6.4.1 Write…………………………………………………………………………….…..33

6.4.2 Hold ……………………………………………………………………………...33

6.4.3 Read ……………………………………………………………………………...33

CHAPTER 7: PERFORMANCE EVALUATION AND COMPARISION OF DIFFERENT SRAM CELLS ....…………………………………………………...35

7.1 Stability of SRAM cells……………………………………………………...35

7.2 Process Variations……………………………………………………………35

7.3 Static Noise Margin curves of SRAM cells……………………………….....37

7.3.1 Hold SNM……………………………………………………………………37

7.3.2 Read SNM ………………………………………………………………….....39

7.3.3 Write SNM …………………………………………………………………...40

7.4 SNM variation with the supply voltage scale-down……………………………43

CHAPTER 8: EFFECT OF PROCESS VARIATION……………………………………45

8.1 Effect on read and write operations……………………………………….....45

8.2 Numerical analysis……………………………………………………………….49

CHAPTER 9: ERROR CORRECTION CODING…………………………………………..52

9.1 HAMMING CODE……………………………………………………………52

9.2 MODIFIED HAMMING SEC-DED CODES ……………………………………57

9.3 COMBINATORIAL REPRESENTATION MATRICES………………………………..57

9.3.1 Optimal odd-weight column codes ……………………………………...57

9.3.2 Construction Procedure……………………………………………………………….58

9.4 PRODUCT CODE ………………………………………………………………..60

9.5 MATRIX CODE……………………………………………………………………..62

CHAPTER 10: CONCLUSIONS AND FUTURE WORK……………………………………..64

APPENDICES………………………………………………………………………………..67

List of Figures

Figure1: Basic SRAM cell …………………………………………………………….…...14

Figure2: Experimentally fabricated SRAM cell size trend…………………………………………17

Figure3: 6T SRAM cell…………………………………………………………………...…18

Figure4: Simulation of 6T SRAM cell……………………………………………………….…19

Figure5: Standard 6T SRAM cell layout…………………………………………………….…..20

Figure6: Modified latch-type voltage sense amplifier………………………………………...21

Figure7: Layout of the pre-charge circuit ……………………………………………………….22

Figure8: NMOS-type write driver circuit………………………………………….………….23

Figure9: Layout of the write driver ……………………………………………….…….……23

Figure10: Normalized read SNM and WNM of a standard 6T SRAM cell for different cell ratios (CR), while pull-up ratio (PR) was fixed to ………………………………………….………24

Figure11: Normalized read SNM and WNM of a standard 6T SRAM cell for different pull-up Ratios (PR), while cell ratio (CR) is were fixed to 2……………………………………………….24

Figure12: Butterfly plot represents the voltage-transfer characteristics of the cross-coupled inverters in the SRAM cell ….……………………………………………………………….…..25

Figure13: The schematic of a 7T SRAM bit cell………………………………….……………….27

Figure14: Simulation of 7T SRAM cell………………………………………...…………..28

Figure15: The schematic for 8T SRAM cell……………………………………...…………...29

Figure16: Simulation of 8T SRAM cell……………………………………………...……...30

Figure17: Schematic representation of a 9T SRAM cell……………………………………….………..31

Figure18: Simulation of 9T SRAM cell………………………………………..………………..32

Figure19: Schematic representation of a 10T SRAM cell……………………………………......33

Figure20: Simulation of 7T SRAM cell……………………………………………….…………34

Figure21: Measurement of read static noise margin (SNM) at VDD=0.9V for 45nm technology node (a) standard 6T SRAM cell, and (b) read SNM free 8T SRAM cell ……………………...….36

Figure22: Measurement of read static noise margin (SNM) at VDD=0.3V for 45nm technology node (a) standard 6T SRAM cell, and (b) read SNM free 8T SRAM cell……………………………36

Figure23: Circuit for hold state SNM measurement……………………………………………..37

Figure24: Hold state SNM curves of 6T to 10T SRAM cells…………….………………………..38

Figure25: Circuit used for measuring the read state SNM of 6T SRAM cell…..………………..…….39

Figure26: Read state SNM curves of 6T to 10T SRAM cells………………..……………………...40

Figure27: Circuit used for measuring the read state SNM of 6T SRAM cell…………………………41

Figure28: Read state SNM curves of 6T to 10T SRAM cells…………………………………………42

Figure29: Hold SNM variation of a 6T SRAM cell, with variation in V_{DD}…………………………...43

Figure30: Hold SNM variation of a 7T SRAM cell, with variation in V_{DD}…………………………...43

Figure31: Hold SNM variation of an 8T SRAM cell, with variation in V_{DD}…………….…………44

Figure32: Hold SNM variation of a 10T SRAM cell, with variation in V_{DD}…………………………44

Figure33: Power dissipation, read time, write time variation of 6T SRAM cell………….……..45

Figure34: Power dissipation, read time, write time variation of 7T SRAM cell……….………..46

Figure35: Power dissipation, read time, write time variation of 8T SRAM cell……….……..47

Figure36: Power dissipation, read time, write time variation of 9T SRAM cell……….……..48

Figure37: Power dissipation, read time, write time variation of 10T SRAM cell……………………48

Figure38: Read time variation on 6T-10T SRAM cells……………………………………........49

Figure39: Read time variation on 6T-10T SRAM cells……………………………………........50

Figure40: Read time variation on 6T-10T SRAM cells…………………………………….………51

Figure41: Hamming Cube…………………………………………………………………...…...52

Figure42: Parity bits generation…………………………………………………………....54

Figure43: Data bits (0100)…………………………………………………………….……...55

Figure44: Parity Bits generated by modulo 2 additions (101)……………………………………...55

Figure45: Data read from the array of SRAMs……………………………………..…………56

Figure46: Syndrome bits……………………………………………………………………………56

Figure47: Parallel Decoding Circuit [30]……………………………………………………..59

Figure48: Array of a product code [31]………………………………………………..………..60

Figure49: Syndrome array for received array [31]…………………………………….…………..61

List of Tables

Table1: Ideal scaling method proposed by Dennard……………………………………..…………15

Table2: Relation between 'xx nm' and ITRS parameter values for 'xx nm' logic CMOS……………16

Table3: Variation of threshold voltage with the increase along the feature scaling……………..........35

Table4: Static Noise Margins of various SRAM cell designs………………………………..…….42

Table5: Mean and standard deviations for the read time variation of all the memory cells…………..49

Table6: Mean and standard deviations for the write time variation of all the memory cells………….50

Table7: Mean and standard deviations for the power consumption of all the memory cells………….51

Table8: Numerical Analysis of different Hamming codes……………………………………………57

Table9: A 16 bit data word with check bits and parity bits……………………………………...62

1. INTRODUCTION

In this project we aim to explore on methods and techniques that can be applied to deduce an overall power and ensuring reliable operation. A memory with fault tolerance and low power can be a key to develop reliable embedded system.

1.1 Aims: The aim of this project is to study a chip's behavior under large process variations with statistical delay assumptions and devise coding techniques of memory devices, to increase the flexibility of the device against process variation and thus to increase the yield.

1.2 Objectives:

- To research methods and techniques for ultra-low voltage operation of memory cell.
- To analyze the performance of existing memory cells, compare the static noise margin (SNM) and failure rate.
- To design and implement new approaches to increase the SRAM performance while reducing its power consumption, such as improving reliability and bit-cell stability, increasing the read and write margins etc.
- To use higher level tools for the evaluation of Error Correction overhead in memory to achieve reliable operation.

1.3 BACKGROUND AND CONTEXT:

With the increasing growth of CMOS technology, millions of transistors can be housed on a single crystal silicon substrate. These advances in Nano-CMOS technology have led to smaller, faster, efficient systems. Embedded SRAMs have taken over CMOS computing chips taking well over half of the total transistor count of high performance ICs [23]. Due to this, designers are forced to decrease the SRAM layout area as much as possible which in turn leads to a high transistor density. As an impact of the high density, the circuits become extremely sensitive to process variations. As the circuit geometry decreases, variation in devices increases because variations do not decrease as rapidly as the feature sizes [7]. These variations are significant in the micron and sub-micron range.

Process variations limit performance gains and stability of circuits which is not desirable, as there is an increasing demand for devices that consume minimum power. Conveyable electronic devices in particular, have a critical requirement of low power consumption in order to stretch battery lifetime. Interest in research on scaled devices have always been high, as a significant reduction in circuit dimensions would enable the use of smaller die areas that would in turn result in reduced costs. The requirement for design robust systems that are resilient to such hardware imperfections have hence become more of a necessity. System-level costs such as energy, performance, power, area, design, and validation costs remain a major challenge, along with achieving an acceptable level of robustness.

Memories, on-chip communications, and embedded computing systems based on the continuously shrinking CMOS processes will therefore need to include fault tolerant (FT) techniques to improve their reliability. Therefore, careful incorporation of fault tolerant techniques in reliable system design is necessary in order to avoid degrading the system speed or increasing the power consumption.

Hence, efficient active and leakage power saving SRAM designs should be explored in order to increase the reliability and lifetime of the battery powered applications. There are mainly three areas with strong potential of power saving: (a) Capacitance reduction of word lines and the number of cells connected to them, data lines, I/O lines and decoders (b)reduce static current by partial activation of multi-divided arrays(both for word-line and bit-line); (c) lowering operating voltage resulting from external power supply reduction and half-Vdd pre-charge.

1.4 THESIS OUTLINE

Chapter 2 presents SRAM architecture, which explains the operation of word-lines, bit-lines, transistors used, row decoder, column decoder and the support circuitry involved in detail.

Chapter 3 embarks on the CMOS semiconductor device modeling, and provides an insight on the evolution of the SRAM cell size trend. The process of SRAM scaling has been explained, and the device models used to simulate the proposed SRAM cells are discussed in detail.

Chapter 4 describes the architecture and functioning of a SRAM cell, by using standard 6T SRAM cell as an example. It analyzes the different modes of operation, simulation waveforms, layout, and the peripheral circuitry used.

Chapter 5 considers the requirement to size an SRAM cell. Specifically, it addresses the trade-offs in SRAM cell design.

Chapter 6 proposes five different SRAM cell topologies, adopted from the papers [13-16]. Then the implementation of these designs and their simulation waveforms are analyzed in detail.

Chapter 7 presents the stability analysis and performance evaluation of these simulated SRAM cells. The effect of process variation on the stability of memory cells is also studied. Hold, read and write static noise margins of an SRAM cell and the corresponding circuits used for SNM measurement are presented. Variation of static noise margin with supply voltage reduction in discussed and the related waveforms are explained.

Chapter 8 summarizes the effect of process variation is along with the waveforms generated through Monte Carlo simulations for stability analysis.

Chapter 9 presents the various algorithms used for the implementation of error correction coding in memory. Implementation of Multiple bit error correction using Hamming codes has been explained along with numerical results. Other error correction techniques such as HSIAO codes, Product codes and Matrix codes are explained and evaluated.

Chapter 10 presents thesis conclusion and future work.

2. SRAM ARCHITECTURE

This section briefs about some elementary facts on aspects of a VLSI system like SRAMs, their categories, process variations, power consumption, reliability and security.

2.1 About SRAMs: "RAM or Random Access Memory is a form of integrated circuits that allow stored data to be accessed in any order with a worst case performance of constant time" [25]. Static RAMs (SRAM) and Dynamic RAMs (DRAM) are the two main forms of Random Access Memories. Dynamic RAMs are comprised of memory cells each of which has a transistor and a capacitor to store a bit of data. The high or low state of the bit is stored by the capacitor, and a control circuitry reads or changes this state, making use of the transistor as a switch. DRAMs require being periodically refreshed as the capacitors tend to discharge over time and the information is lost. Due to the periodic refreshing being a necessity, they are called 'dynamic' RAMs. Static RAMs are relatively complex as they generally have four to six transistors. As a result of which, this form of RAM is more expensive to produce, much faster, and less power consuming when compared to DRAMs. Static RAMs make use of a bi-stable latching circuitry to store each bit. They are known to exhibit data reminisce, (Residual representation of data that remains even after attempts have been made to remove or erase the data [25]) but the data can be lost if the memory is not powered. SRAMs are generally used in the caches of modern computer for faster data accessing.

2.2 Types of SRAM: SRAMs are classified based on their functionality, type of transistors used, and memory size.

<u>Functioning</u>: There are two types: synchronous and asynchronous. Asynchronous SRAM has a sequential pattern of READ and WRITE operation and is independent of clock frequency; while the synchronous SRAM has an overlapping READ and WRITE operation, initiated by the clock edges.

<u>Type of transistor used</u>: CMOS (complementary metal oxide semiconductor) and BJT (bi-polar junction transistors). The bipolar type consumes a lot a power and is very fast. However, it has become obsolete now.

<u>Memory Size</u>: Single density, DDR (Double data rate), and Quad data rate SRAM

2.3 Power Consumption: Two factors decide the power consumption in a CMOS circuit: Static power consumption and Dynamic power consumption. Static power consumption occurs when the circuit is not charged and the inputs are held at a certain logic level. It is generally very low in CMOS circuits owing to leakage currents. Dynamic power consumption occurs when switching takes place at high frequency and when the output capacitive load is continuously charged and discharged. Dynamic Power Consumption is significantly high and constitutes a larger part of the overall power consumption.

2.4 Reliability: "Its defined as the conditional probability that a system provides continuous proper service for a given interval of time, considering its proper functionality at the beginning of the interval" [32] CMOS reliability is affected by a number of physical failure mechanisms, owing to the variation in temperature, voltage, current density, etc. Furthermore, with the development of CMOS technology, the area per bit has been scaled down and more and more transistors are fabricated in a die, which in turn scales down the supply voltage. This results in the node charge reduction linearly with the channel length. "As early as 1962 [Chinese dissertation], it was pointed out that, if channel length scales below 1μm, even a single cosmic ray particle strike would short-circuit the source and potentially makes some temporal changes in the circuit." Such errors were called soft errors.

2.5 Security: Growing demand for information security in the recent years has created a restraining case for efforts to build secure electronic systems. Since embedded systems are universally used to handle sensitive data, they put across several security challenges. Areas involving intensive research in security include cryptography, computing and networking.

2.6 Process Variations: Process variation can be defined as the alteration in the attributes of transistors (Threshold voltage, gate oxide thickness, geometry of the device, doping concentration, patterning proximity effect etc.) during the fabrication process of integrated circuits. Process variations in recent CMOS technology have been major intimidating factors for SRAM cell robustness. Closely matched transistors inside memory cells yield good noise margins. The fact that the ratio of these variations within a device not being constant throughout the size of the device, makes it more challenging to predict and minimize the effects of process variation.

Primary architecture of a static RAM is comprised of one or more rectangular arrays of memory cells with a few additional backing circuitries for decoding addresses, and implementing the required "READ and WRITE" operations. SRAM memory arrays are placed in the form of a matrix, i.e., rows and columns of memory cells called word-lines and bit-lines, respectively. The memory array consists of 2n words of 2m bits each. Each bit of information is stored in one memory cell. SRAMs that are manufactured by IBM contain word-lines that are built from poly-silicon and the bit-lines that are from metal. The points where the bit-lines and word-lines crossover, represents a unique location address for each memory cell. The number of arrays on a memory chip is dependent on the memory size, operation speed, data size and testing requirements. Therefore, large sized memories can be replicated into different memory blocks with smaller number of rows and columns. A basic SRAM cell is shown in Figure 1.
M3, M1 and M4, M2 are two cross coupled inverters, M5 and M6 are access transistors. The access transistors are connected to the word-lines at their respective gate terminals, and the bit-lines at their source/drain terminals. The function of a word-line is to select a particular cell; whereas that of a bit-line is to read from or write into a cell. Internally, the cell holds the stored value on one side and its complement on the other side. The two complementary bit-lines are used to improve speed and noise rejection properties [1]. Depending on the address inputs, one of the output lines of the row decoder goes high, which selects the access transistors corresponding to a particular row. All the memory cells in that row are selected and one cell in each of the columns is selected, which is connected to the bit-line. As seen from the figure, there are two data lines; BL and BL' complementary to each other because of the inverter configuration. Cells in the columns are selected based on the inputs from the block decoder.

Hence, depending on the input address, the combination of the row and block decoder inputs, a single memory cell is selected for information transfer. As the memory size gets larger, word-lines and bit-lines get extremely long. Also, since they have large capacitances and resistances, a great deal of delay is caused in charging and discharging these lines.

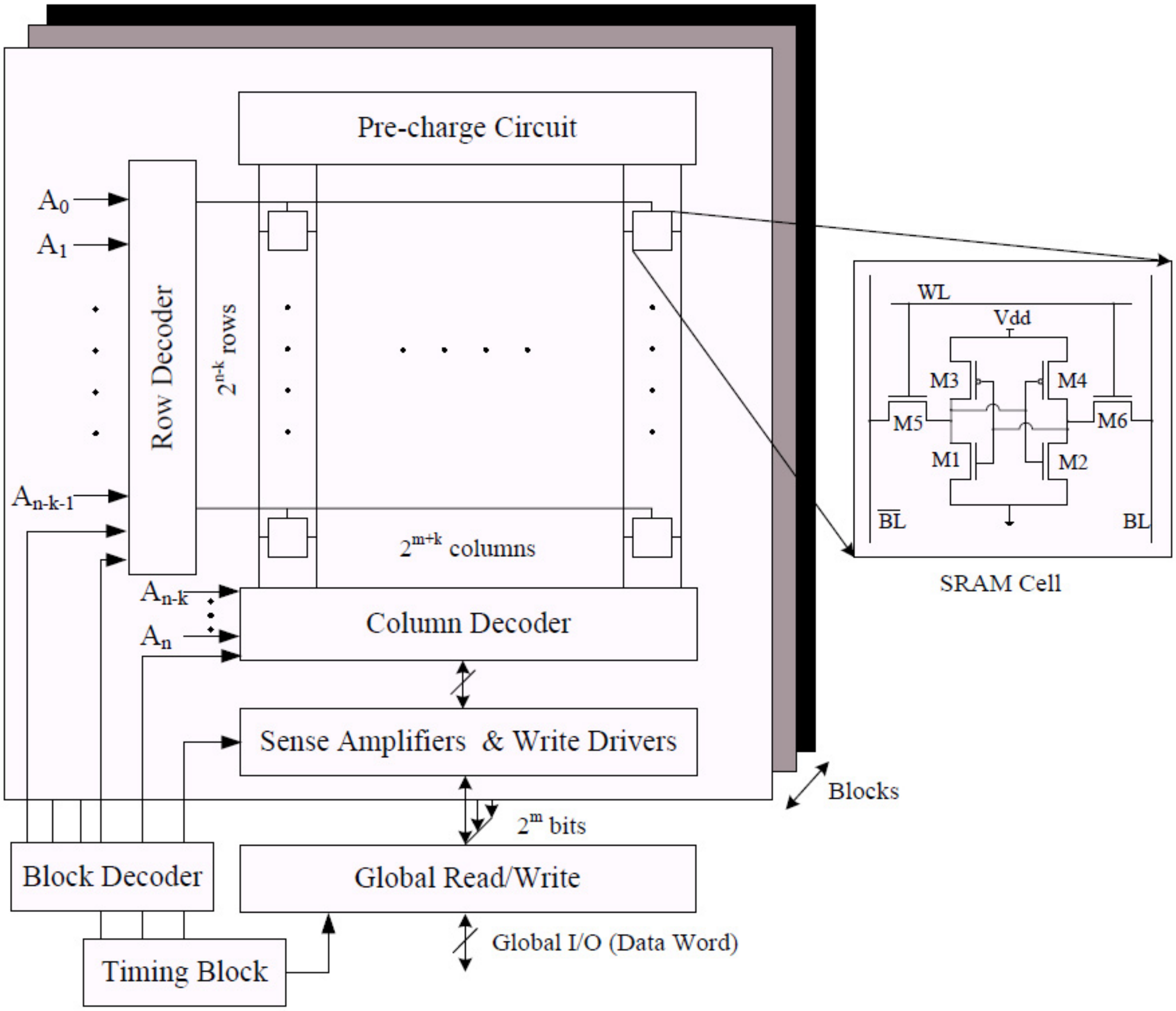


Figure1: Basic SRAM cell

2.7 Support Circuitry:

The additional circuitry of the memory chip facilitates the user to read the data stored in the memory's cells, and write data to the cells. This circuitry is generally comprised of:

• Address logic for the selection of rows and columns.
• Translation logic to read the data from a memory cell and to transfer that data to the data I/O.
• Write logic to collect the data supplied by the user at the input and to store it in a memory cell.
• Output enables logic for the prevention of data from emerging at the outputs unless explicitly specified.
• Internal counters and registers for keeping record of the burst address sequences, pipelined data, and other control activities on the chip.
• Clock circuitry to regulate the timing of the various read and write operations. [24]

2.8 Embedded SRAM memory cells:

Embedded memory is an integrated on-chip memory. It is highly acclaimed of late because of the extensive usage of high integration systems. Conveyable electronic devices, which have a critical requirement of low power consumption, need very large amount of on-chip memories. To render to this huge requirement of memory, a mainstream technology known as System in Package (SiP), is being used, where, more than one active electronic component with other passive and optical components are enclosed in a single package or module. An alternative solution is to accommodate memory along with other logical components, on the same die.

3. DEVICE MODELS

3.1 Semiconductor device modeling: A brief overview of semiconductor device modeling is presented in this chapter which illustrates the mechanism of representing solid state devices. "**Semiconductor device modeling** is the process of creating models for the electrical behavior of devices, based on their fundamental physics such as doping profiles." [33]. SPICE uses detailed and accurate device models to define individual elements, comprising the device, in various versions. Making use of standard models helps a great deal to reduce the amount of time required to simulate a circuit design. Each element, within a netlist that invokes a model is known as an *instance* of that model. Device models are proven to be most useful in circuit level power estimation and they are the most accurate power models.

The growth of MOS large scale integration is marked by the sizing down of the comprising components such as MOSFET's from their initial development stage in the early 1970's. The key to achieve high performance and low power consumption is down-scaling, and hence the technique of reducing the gate length has come into practice. The ideal scaling method proposed by Dennard shown in Table 1 [34] enhances the performance without any major increase in power consumption, provided the chip area is maintained constant.

Ideal down-scaling scheme.

Geometry and supply voltage	L_g, W_g, T_{ox}, V_{dd}	K	Scaling K: $K = 0.7$ for example
Drive current in saturation	I_d	K	$I_d = V_{sat} W_g C_o (V_g - V_{th})$ C_o: gateCper unit area $\rightarrow W_g (t_{ox}^{-1})(V_g - V_{th}) = W_g t_{ox}^{-1} (V_g - V_{th}) = KK^{-1}K = K$
I_d per unit W_g	I_d/µm	1	I_d per unit $W_g = I_d/W_g = 1$
Gate capacitance	C_g	K	$C_g = \varepsilon_o \varepsilon_{ox} L_g W_g / t_{ox} \rightarrow KK/K = K$
Switching speed	τ	K	$\tau = C_g V_{dd} / I_d \rightarrow KK/K = K$
Clock frequency	f	$1/K$	$f = 1/\tau = 1/K$
Chip area	A_{chip}	α	α: Scaling factor $\rightarrow$ In the past, $\alpha > 1$ for most cases
Integration (# of Tr)	N	α/K^2	$N \rightarrow \alpha/K^2 = 1/K^2$, when $\alpha = 1$
Power per chip	P	α	$fNCV^2/2 \rightarrow K^{-1}(\alpha K^{-2})K(K^1)^2 = \alpha = 1$, when $\alpha = 1$

Table1: Ideal scaling method proposed by Dennard [34]

“Table2 shows typical technology generations for logic LSIs since early 1970s towards future expected limit.” [34]

'xx nm' CMOS technology			ITRS (2008 Update)		
Technology node (nm)	Starting Year		Year	Half pitch (1st metal) (nm)	Physical gate length (nm)
Commercial logic CMOS products for high-performance logic					
45	2007	↔	2007	68	32
			2008	59	29
32	2009?	↔	2009	52	27
			2010	45	24
22	2011?–2012?	↔	2011	40	22
			2012	36	20
16	2013?–2014?	↔	2013	32	18
			2014	29	16

Table2: Relation between ‘xx nm’ and ITRS parameter values for ‘xx nm’ logic CMOS. [34]

‘xx’ nm or um was used to represent the lithography resolution, which was the half pitch of the lines, the minimum gate length and the metal line width. [provide reference] ‘xx’ value has been decreasing over the years at an approximate rate of 0.7 for every 3 years on an average. Although ‘xx’ nm had nothing to do with the physical gate length and was just a commercial name as in the case of 45 and 32 nm nodes (Table), it happens to be close to the physical gate length from the 22nm node. Till now, half pitch of the lithography has been much larger, when compared to the gate length of the logic CMOS. However, due to the recently developing trend of physical gate-length reduction anticipated by the recent versions of the ITRS has been very disruptive for the semiconductor industries to keep pace with. Semiconductor companies except Intel have considerably reduced their investment towards research and development, owing to the recent recession in economy. Hence, there is a potential possibility that the reduction in gate length is delayed further.

3.2 SRAM scaling:

SRAM is an essential part of logic devices, as cache memories are comprised primarily of SRAM’s and the area that they consume is considerably large. Decreasing the gate length and supply voltage of an SRAM cell is challenging because, even a small off-leakage current in a single transistor in an SRAM cell results in a massive off-leakage current in the entire chip. Therefore, the gate length and supply voltage used within the SRAM cell is generally designed to be much greater than that used in the functional part of the chip. However, the fabrication results of the SRAM cell show that there is a reduction rate of 1/2 to 2/3 for every generation until the 22nm node [34]. New techniques have been introduced to realize 32nm and 22nm SRAM cells. The process of double lithography is employed to realize square endcap of the gate pattern and to reduce the supply voltage variation high-k/metal gate stack is used.

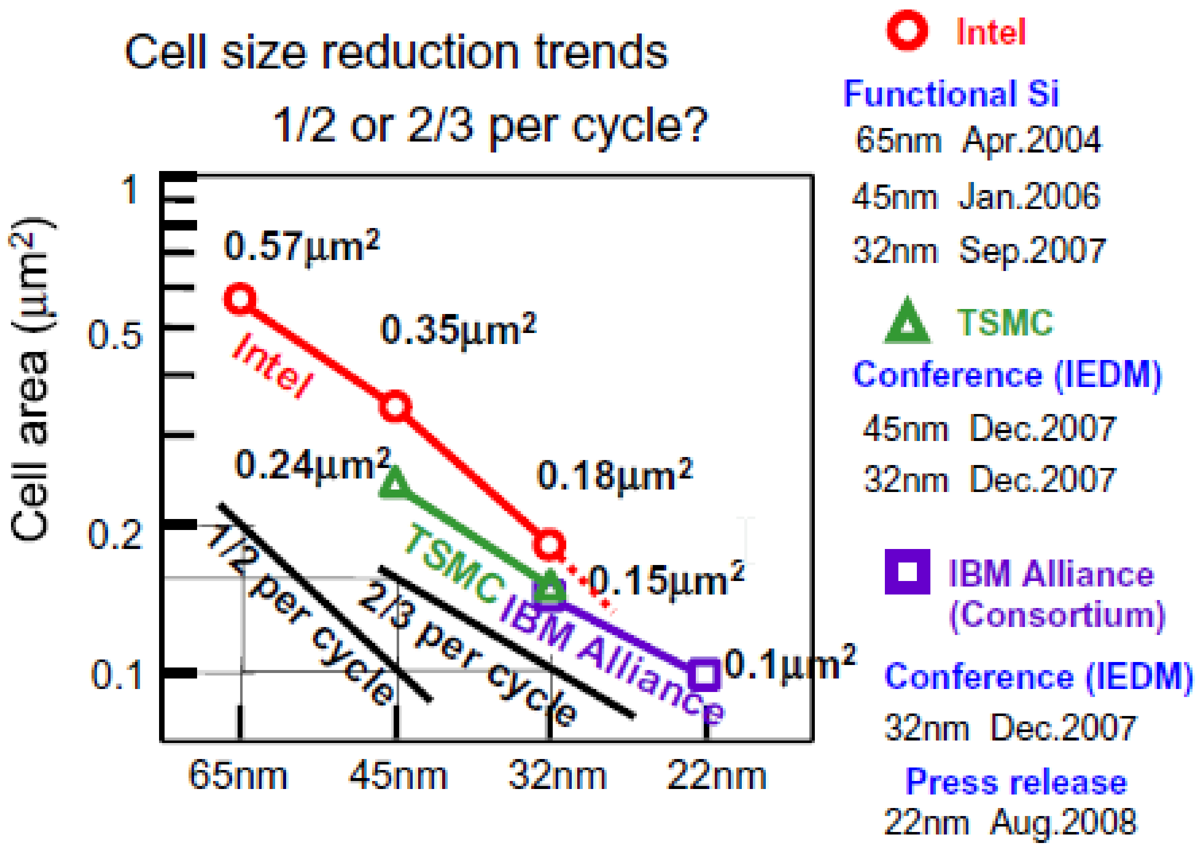


Figure2: Experimentally fabricated SRAM cell size trend [34]

All SRAM cells (6T to 10T) have been simulated in the following three device technologies, adopted from the predictive technology model website:

22nm PTM model for metal gate/high-k CMOS
32nm PTM model for metal gate/high-k CMOS
45nm PTM model for metal gate/high-k CMOS

These are level 54 BSIM4.0 (Berkeley Short-channel IGFET Model) models. BSIM models consider the variation of model parameters as a function of sensitivity of the geometric parameters, for precision modeling of integrated circuits. The idea was to study, evaluate and compare the performance of the SRAM cells under the effect of process variations, with the use of different device models.

4. STANDARD SIX TRANSISTOR (6T) SRAM

4.1 6T SRAM cell architecture: Two cross-coupled CMOS inverters together constitute the core of the cell. The output voltage of each inverter is fed as the input to the other. This feedback mechanism provides stability to the inverters to maintain their respective states. These inverters serve as storage elements. The structure is designed to be symmetrical to facilitate differential signaling, which makes even the minute signal variations very easy to detect.

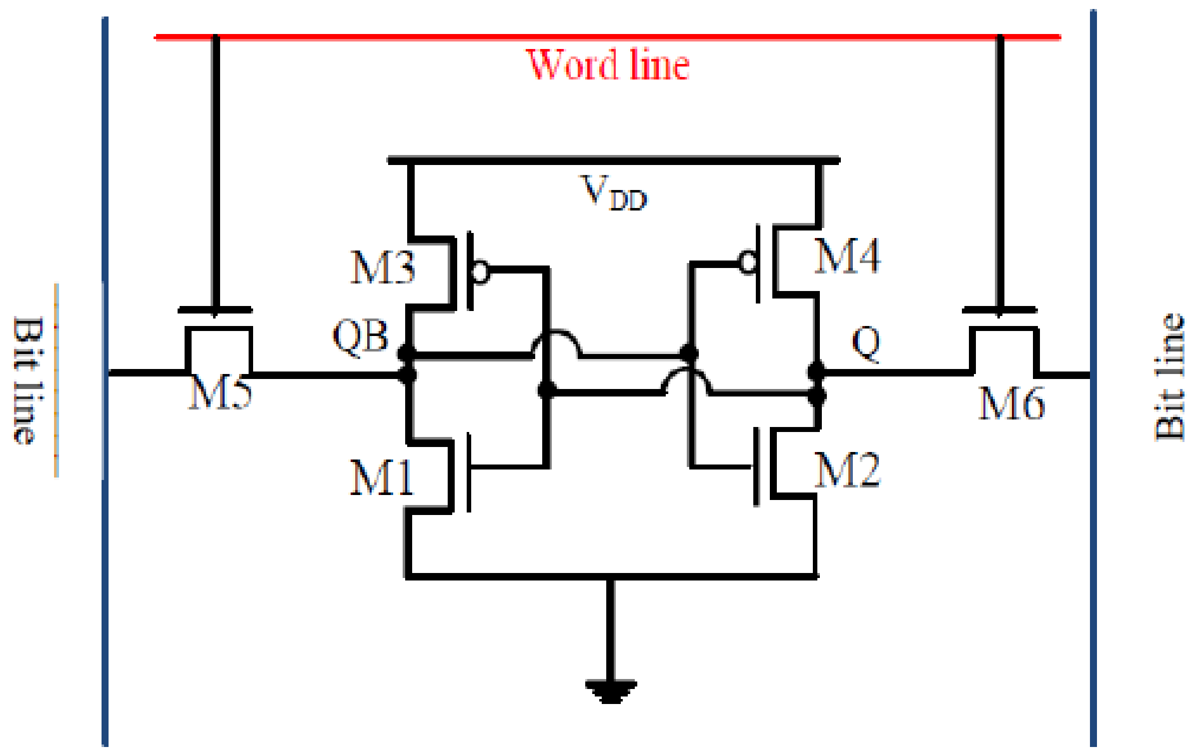


Figure3: 6T SRAM cell

An SRAM cell has three different states. *Standby*, implying that the circuit is idle, *Reading*, which means that the data has been requested, and *writing*, indicating that the circuit is updating the contents.

4.2 Standby mode:

In case of the word line not being asserted, bit lines are disconnected from the cell by the *access* transistors M5 and M6. The cross-coupled inverters, M1 – M4 will still continue to back up each other, unless they are disconnected from the supply voltage.

4.3 Reading mode:

If the memory has a **1** stored at Q, the read cycle starts by the pre-charging of both the bit lines to a logical **1**, followed by the assertion of the word line WL, which enables the *access* transistors. The values are then stored in Q and Q', and are transferred to the bit lines by letting BL retain its pre-charged value and discharging BL' down to a logic zero via the transistors M1 and M5. Transistors M4 and M6 pull the bit line to VDD (logic 1). On the contrary, if the memory has a **0** stored, BL' would be pulled to logic **1**and BL would be discharged down to logic **0**. Delta, the small difference between BL and BL' would then be amplified and sensed by a *sense amplifier* which identifies the line with the higher voltage, thus determining the content of the memory. The speed of the read operation is directly proportional to the sensitivity of the sense amplifier.

4.4 Writing mode:

The bit lines are pre-charged to the value to be written. That is, if logic 0 has to be written, BL' is pre-charged to **1** and BL to **0**. Assertion of WL takes place followed by the latching of the value to be stored. Overriding of the previous state of the cross coupled inverters is possible because, the bit-line input drivers are designed to be stronger than the transistors within the cell. Transistors in an SRAM cell require careful sizing in order to ensure proper operation.

The graphs obtained from the simulation of the 6T SRAM cell are shown in the figure below. It shows the output waveforms at BL, BL', Q and QB. To study read and write operation in detail, a one-bit cell has been designed along with a write driver, pre-charge circuit and a sense amplifier.

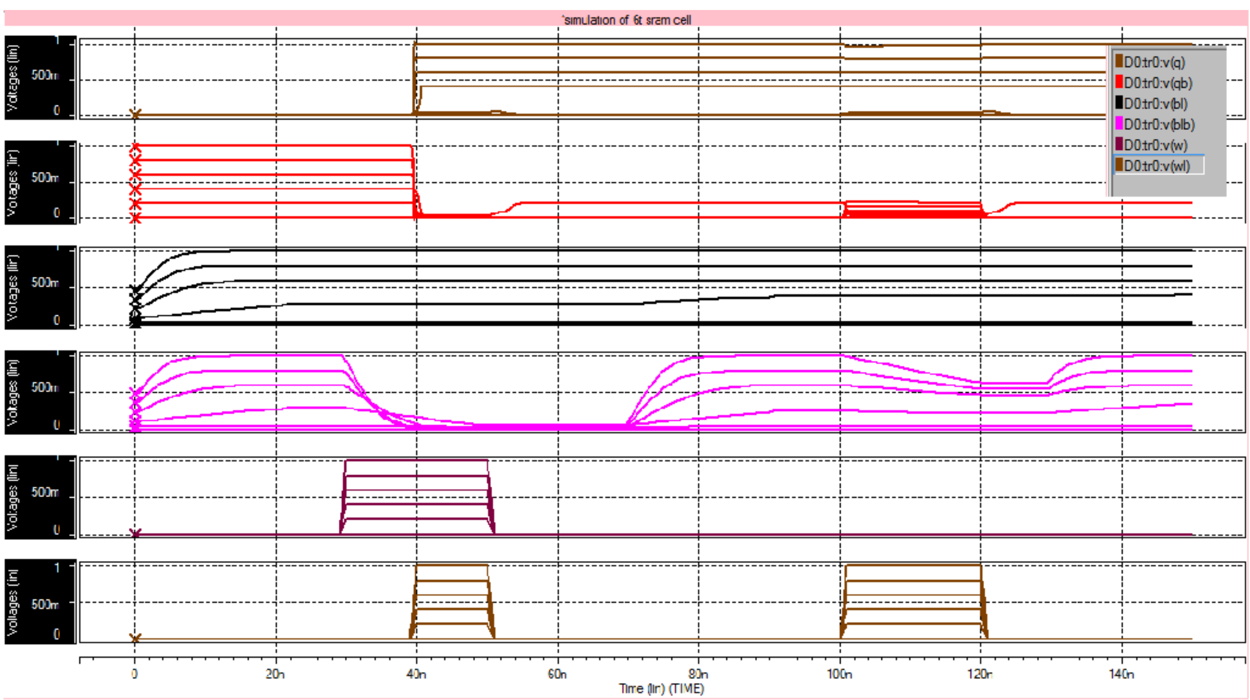


Figure4: Simulation of 6T SRAM cell

PC signal is used to pre-charge the bit-lines to the supply voltage value. It is made high only when the bit-lines are not connected to the memory cell. The data to be written into the memory is represented by the signal 'd'. WE signal is used to enable the write driver circuit. WL is set high just before the beginning of a write operation, when both the bit-lines are pre-charged to V_{DD}. Sense Amplifier Enable (SAE) is used to drive the sensing operation. Pulsed voltage sources are used to trigger these inputs.

4.5 Layout of a standard 6T SRAM cell:

Increasing limitations in lithography, design and process technology, and the mechanisms which lead to mismatches in device parameters, can be well understood by building the layout of the standard 6T SRAM cell. Memory constitutes an important part of digital electronics. In low power designs, suppressing the leakage currents is a primary requirement. By reducing the supply voltage V_{DD}, significant reduction of the leakage currents can be achieved. This reduced supply voltage is termed as the Data Retention Voltage (DRV). IT defines the minimum supply voltage under which the data in the memory is preserved. There has been an effective evolution in the 6T SRAM cell topology owing to the improving technology. Structure complexity and density of memory array are defined by the layout of an SRAM cell. These parameters are of utmost importance in designing a system on chip (SoC) which contains large arrays of memory cells.

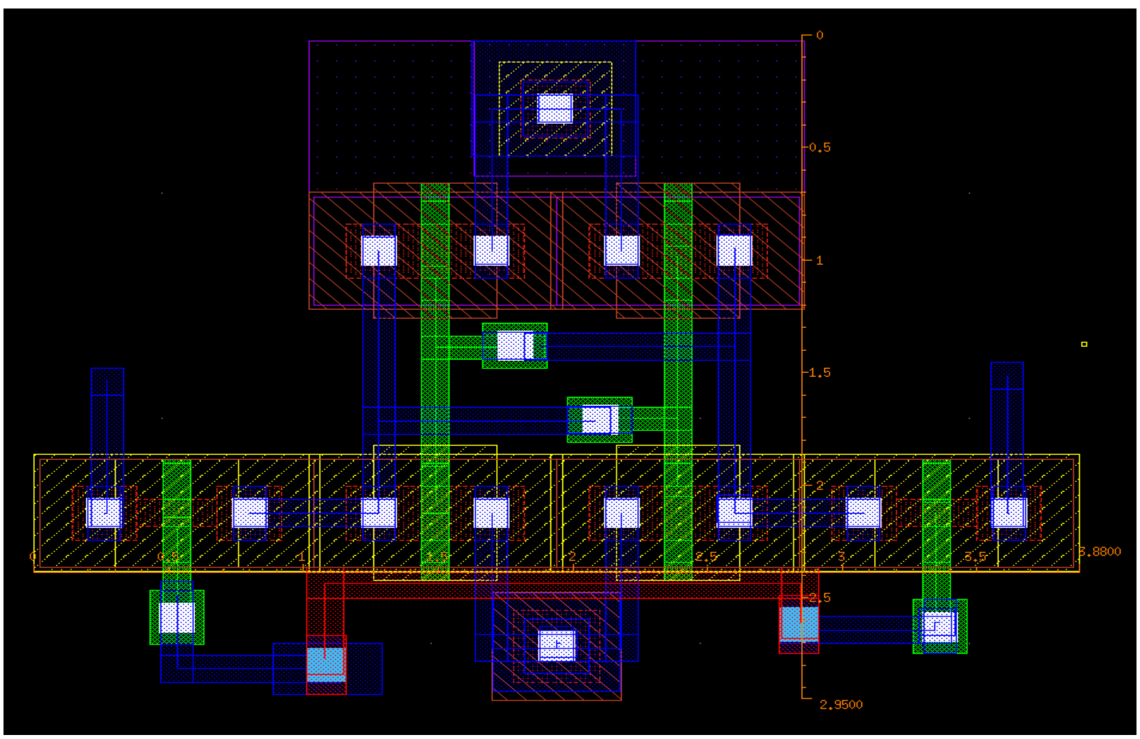


Figure5: Standard 6T SRAM cell layout [2]

Figure shows the layout of a conventional 6T SRAM cell, drawn using 90nm technology. Transistors used are of minimum feature size. It can be seen that the cell structure is symmetrical and the layout is quite easy to be implemented.

4.6 Peripheral Circuitry:

In order to simulate the read, write and hold states of the memory cell operation, we require a few additional support circuits, which facilitate the proper simulation of these operations.

4.7 Sense Amplifier and bit-line pre-charge:

Sense amplifier is a part of the read circuitry. It is mainly used when the data is to be read from a memory cell. The main function of the sense amplifier is to detect and amplify the small voltage difference from the bitline which holds the data bit (logic 0 or logic 1), which reduces the time required for the read operation. Generally, the sense amplifier circuit is quite simple which usually consists of 2 to 6 transistors in a tiny area of the integrated circuit.

The conventional latch type sense amplifier has two nodes, for input and output terminals. Here, the bitlines are discharged during the decision phase, as they are not directly connected to the amplifier. In this project a modified version of the latch-type sense amplifier is used. The latch circuit is controlled by M5 and M6 transistors and the bitlines BL and BL' are connected to the inputs of these two transistors. M5 and M6 transistors amplify the small difference in current, which is converted to a full swing voltage signal at the output nodes SO and SON [quote reference].

The sense amplifier enable signal has to be driven on time because; the sensing process cannot be recovered once it begins until the circuit is brought back to the meta-stable point where the voltages at the two output nodes are equal.

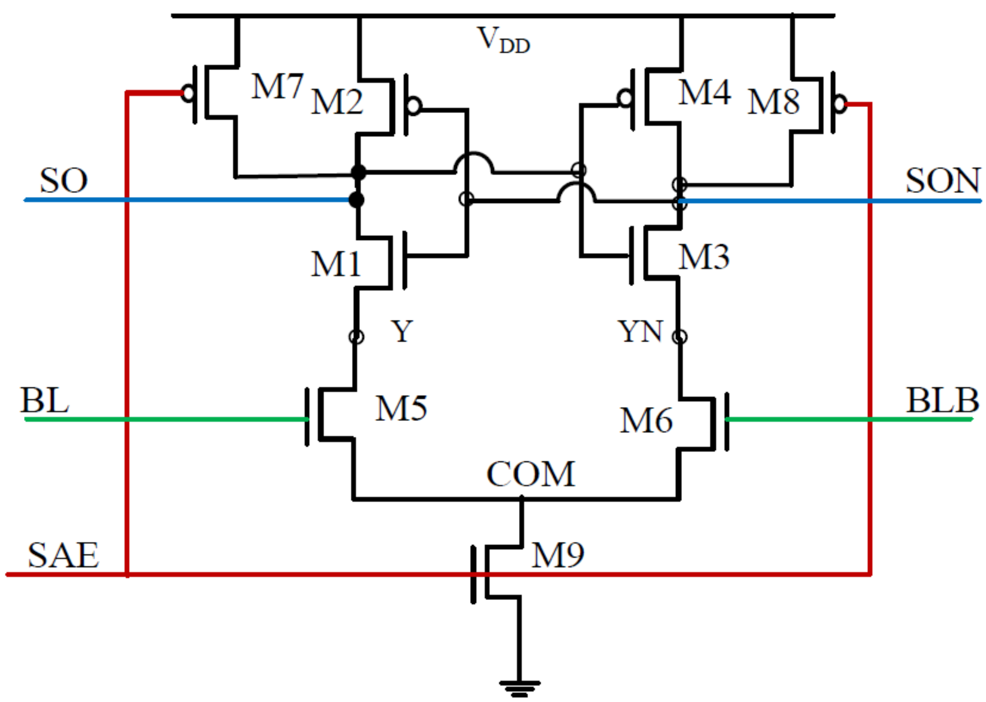


Figure6: Modified latch-type voltage sense amplifier

M7 and M8 are kept on during the OFF state, to keep the output nodes SO and SON at logic 1. Once the sense amplifier enable (SAE) signal is driven, transistor m9 is enabled and the sensing operation starts. When transistors M5 and M6 are on, the voltage at node 'COM' is pulled down to zero. Difference in voltage between the transistors M5 and M6 triggers the discharging of nodes SO and SON. Once the output voltage reaches V_{DD}-V_{tp}, transistors M2 and M4 are turned ON. Following this, one of the NMOS transistors comprising the latch circuit is turned off. As soon as the sensing operation is completed, the current flow cuts off and hence there is no static power dissipation.

4.8 Word Pre-charge:

The word pre-charge circuit is used to pull the bit lines to logic '1' during the read operation. The circuit is built in such a way that there is no conflict with the NMOS transistor in the cross-coupled inverters so that the output node is not affected during the read operation. Input to the pre-charge circuit is given just before the beginning of a read operation. The PMOS transistor goes off whenever the read enable goes high.

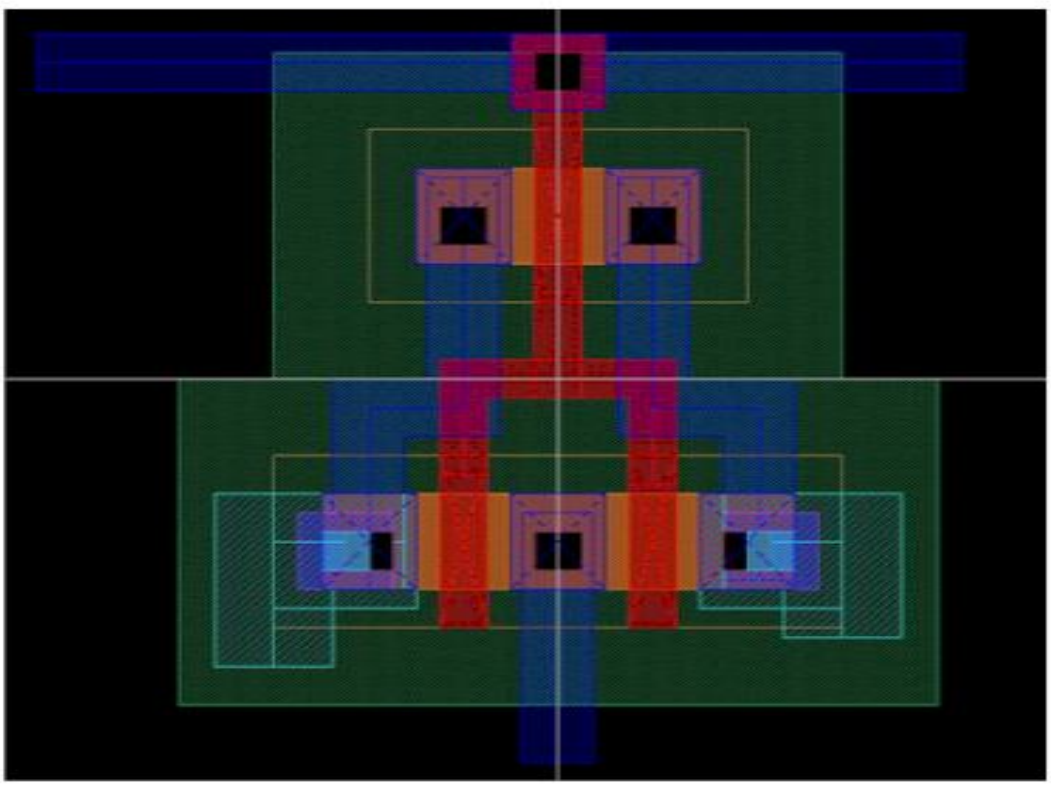


Figure7: Layout of the pre-charge circuit [26]

4.9 Write Driver:

Write driver is used to pull-down the bit-lines from V_{DD} to zero once the data to be written into the cell is known. Once the write driver is ready to function, it enables the word-line. The write driver used here is an NMOS combined driver since NMOS passes a good zero, it can efficiently perform the discharge operation. If the data to be written is logic '1', the bit line is pulled up to logic '1' through the transistor M2. Bit-line' discharges to zero since the data signal is directly connected to transistor M1.

Figure shows the write driver circuit. Only one write driver for each column in the memory array is sufficient to speed up the write operation.

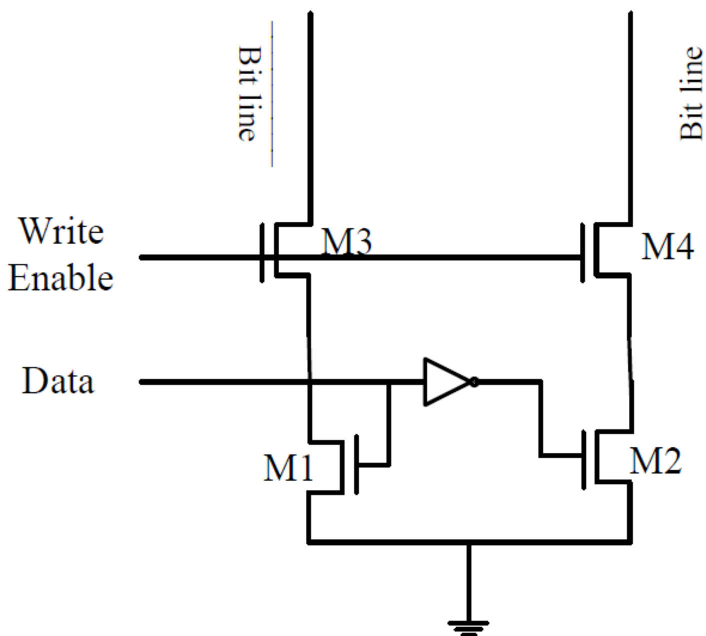


Figure8: NMOS-type WRITE driver circuit

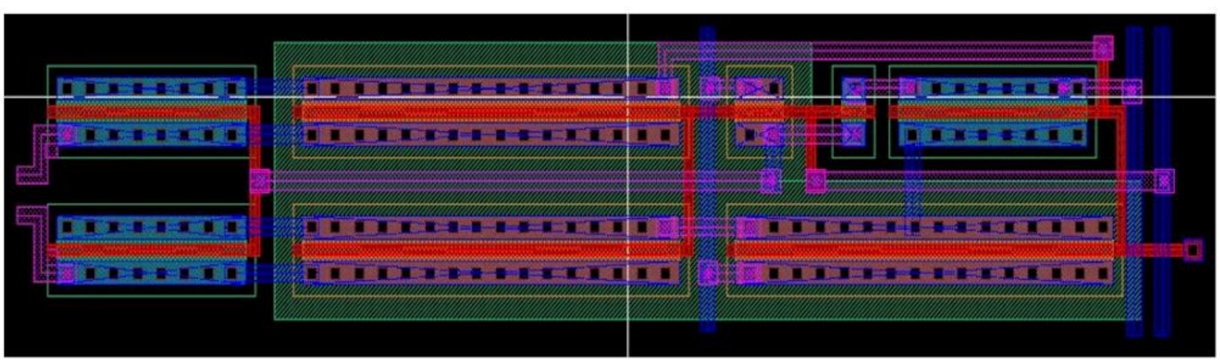


Figure9: Layout of the write driver [26]

5. SIZING REQUIREMENTS IN SRAM CELL

5.1 Sizing and its effects: Due to lowering the supply voltage, there is a continuous diminution in device dimensions in the standard 6T SRAM cell, in an attempt to accomplish the high density and high performance demand of on-chip caches. In a 6T cell, both read and write operations are performed using the same pass-gate devices M5 and M6 as explained previously. Due to this collision of read and write requirements, there is a reduction in noise margins, because of which, the stability of SRAM cell is affected. Hence, for an improved read stability, transistors M1 and M2, which constitute the pull down part of the storage inverters, need to be designed in such a way that they are more substantial than the pass gate devices M5 and M6. Whereas, a contrasted arrangement is favorable for the write operation, that is pass-gate devices, M5 and M6, require being stronger than the pull up devices M3 and M4. The above constraints can be combined to get the following relation. "strength (PMOS pull-up) < strength (NMOS access) < strength (NMOS pull-down)" [1]

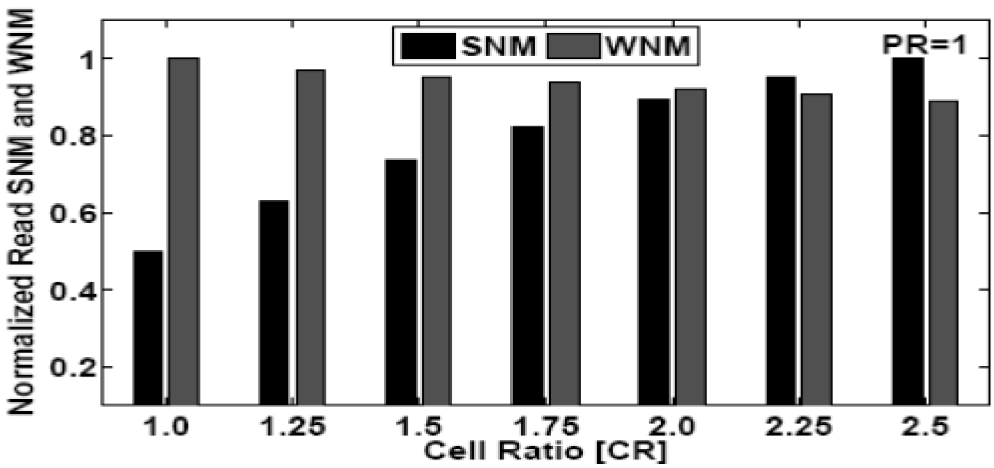
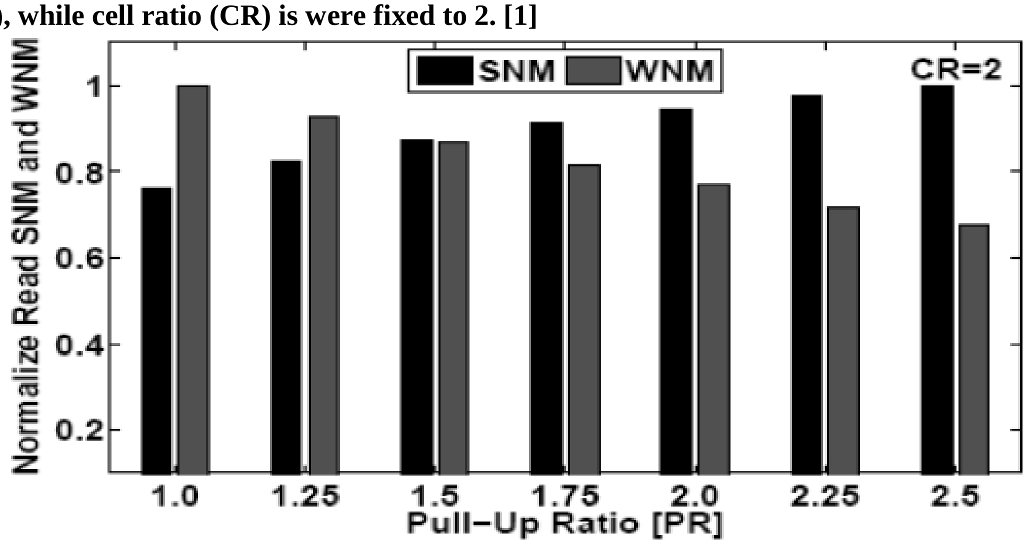


Figure11: Normalized read SNM and WNM of a standard 6T SRAM cell for different pull-up Ratios (PR), while cell ratio (CR) is were fixed to 2. [1]



Normalized read SNM and WNM of a standard 6T SRAM cell for different pull-upratios (PR), while cell ratio (CR) is was fixed to 2. [1]

Increase in SNM, with the increase in cell ratio can be observed by simulating read SNM and write noise margin (WNM) for various cell ratios and pull up ratios. Figure below shows the normalized read SNM and WNM of standard 6T SRAM cell, measured for different cell ratios (CR), keeping the pull-up constant (PR=1). SNM sharply increases with the increase in cell ration, whereas a gradual decrease is seen in WNM. In the next figure, the cell ratio is maintained a constant at 2 and the similar trend is observed again. In general, for high density SRAM arrays, pull-up ration is maintained at 1, while the cell ratio is varied from 1.25 to 2.5, to achieve the minimum size objective.

5.2 Tradeoffs in SRAM design:

5.2.1 Area vs. yield

A memory array is mainly characterized by its functionality and density. Yield is directly biased by the area efficiency and reliable printing of the SRAM cell, which are in turn dependent on the lithography technology. By providing sufficiently large design margins, functionality for large memory arrays against lithography challenges can be ensured. Channel widths and lengths, supply voltage, and transistor threshold voltages determine the device margins. Increasing the transistor sizing increases the noise margin, nut it also increases the cell area resulting in the lowering of density.

5.2.2 Hold margin

When the SRAM is in standby mode, though the memory is not being accessed, the state has to be preserved. The PMOS load transistor holds the stored 1 bit which must be strong enough to counter balance for the sub-threshold and gate leakage currents of all the NMOS transistors branching from the storage node VL. Hold margin determines the minimum supply voltage or the delta retention voltage in standby mode. Designing low power robust memory arrays becomes increasingly difficult at low voltages if hold margin is reduced. In standby mode, the cell Static Noise Margin (SNM) is generally used to assess the hold stability, with the voltage on word line set to zero. “The SNM of an SRAM cell represents the minimum DC-voltage disturbance necessary to upset the cell state, and can be quantified by the length of the side of the maximum square that can fit inside the lobes of the butterfly plot formed by the transfer characteristics of the cross-coupled inverters (Figure).” [1]

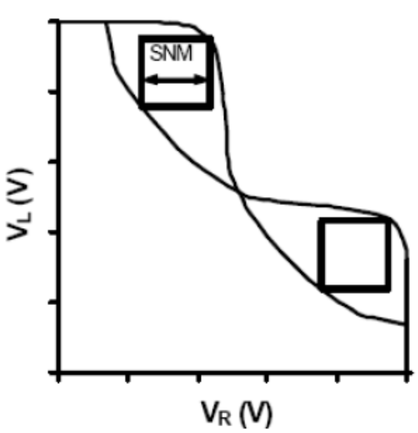


Figure12: Butterfly plot represents the voltage-transfer characteristics of the cross-coupled inverters in the SRAM cell [1]

5.2.3 Read margin:

A read operation requires the bit lines to be in their pre-charged state and the word line to be asserted, so that the storage node voltage VR is at a value which is well above 0V. The resistive voltage divider comprised of the access transistor and the pull down transistor, between BL and ground, determines the value of specified voltage. The cell β-ratio, which is the width/length ratio of the pull-down transistor and the access transistor, determines the maximum value of the storage node voltage, VR. The cell SNM quantifies the read stability during a read access. A decrease in the gain of the inverter transfer characteristics results in a decrease in the separation between the butterfly curves, which implies the reduction in the SNM. Hence, during the read access, the cell is extremely sensitive to electrical disturbs. The read margin can be increased by increasing the sizing of the pull-down transistor, which results in larger area, and/or increasing the gate length of the access transistor, which in turn increases the WL delay and also affects the write margin [1] Additionally, the SNM is degraded due to process induced variations, which results in the instability of the memory cell. The circuit design techniques developed to overcome this problem, have been proven to be insufficient.

5.2.4 Write margin:

Supposing if the value to be written is logic 1, BL and BL' are pre-charged to VDD and 0 V respectively, and then the word line is asserted to VDD to store the new bit. Transistors in the SRAM cell are to be meticulously sized, in order to ensure proper write operation. The maximum BL' voltage that can reverse the cell state, while the voltage on BL is constantly maintained at VDD is termed as the write margin. Increasing the W/L ratio of the access transistor and keeping the size of the pull up device to a minimum, improves the write margin at the cost of the cell area and the cell read margin.

5.2.5 Access time:

The cell access time is defined as the short period of time during which the WL voltage is allowed to be high during any read/write access. If within this time, the operation is not successfully carried out, it results in a read/write access failure. Wire delays and memory array column height determine the access time. Segmentation of the memory into smaller blocks is a popularly employed method to reduce the access time. As a consequence of reducing the column height, the sense amplifier area can become large.

5.3 SRAM bit cell topologies:

Standard 6T SRAM cell has been most popularly used in the manufacture of on-chip caches and microprocessors. However, researches have rendered several SRAM bit cell topologies for different requirements such as smaller memory cell area, minimum dissipation of static and dynamic power, improved static noise margin (SNM) and write ability margin (WAM) and higher performance. The tradeoff among these design metrics has been a primary concern. In the following section, we look at different memory cells under Nano-CMOS technology and evaluate them against Nano-scale process variations.

6. <u>SRAM CELL IMPLEMENTATION</u>

6.1 7T SRAM Memory Cell:

Figure shows the schematic of a 7T SRAM bit cell designed to reduce write power consumption. As the name suggests, this cell is comprised of 7 transistors and the functionality marginally differs from that of a conventional 6T SRAM cell. During the write operation, the feedback path between the two cross-coupled inverters is cut off using an extra transistor M5 [13].

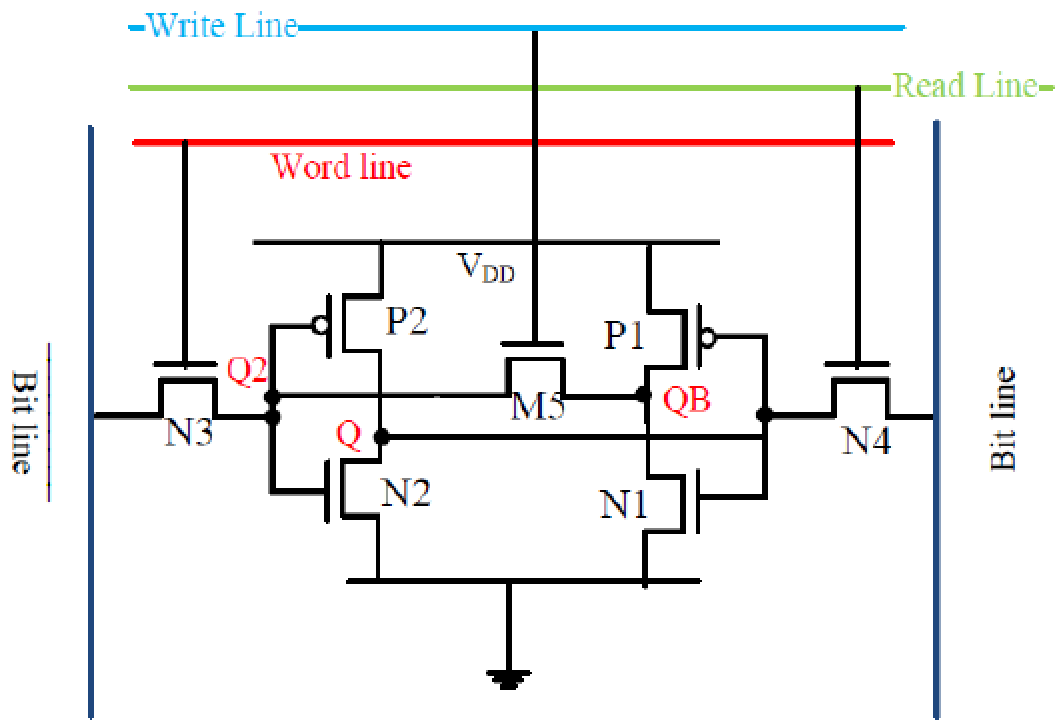


Figure13: The schematic of a 7T SRAM bit cell

6.1.1 Write:

Before the write cycle is begun, M5 is disabled and the feedback path between the two cross-coupled inverters is cut-off. The write operation now only involves BL', therefore, BL' is set to the complement of the value to be written. This is then followed by the assertion of the word line which turns the transistor N3 on, and then the complement value is transferred on to Q2 which drives P2 and N2 together to replicate the data in QB. The memory cell is then disconnected from 'BL' by discharging the word line. Feedback between the two inverters is then reestablished by enabling M5.

6.1.2 Hold:

The memory cell is disconnected from bit lines by disabling word line, read line and write line. The two inverters augment each other to retain the data stored in the data node, without any voltage drop between Q2 and QB.

6.1.3 Read:

The read operations of 6T and 7T SRAMS are identical. The only difference being, 7T SRAM takes a longer time to read since in this case, the critical read path involves the transistors N1, N5 and N3.

The graphs obtained from the simulation of the 7T SRAM cell are shown in the figure below. It shows the output waveforms at BL, BL', WL, W, R, Data, Q, QB and Q2.

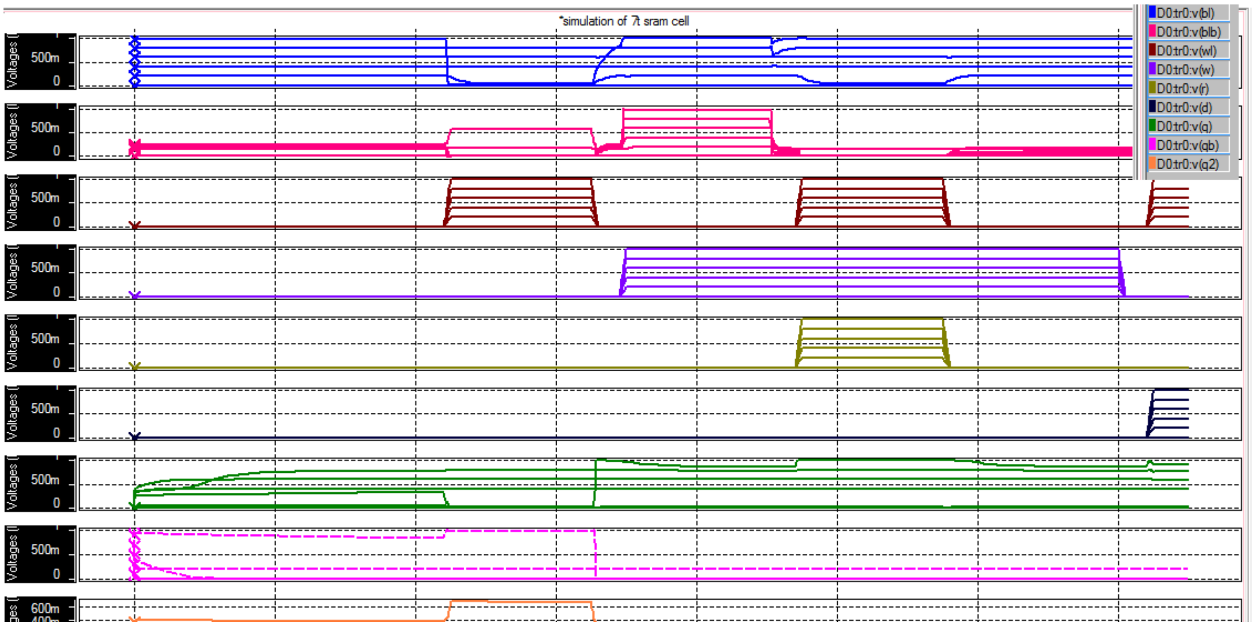


Figure14: Simulation of 7T SRAM cell

7T SRAM cell uses two additional input signals for the read and write operation. WL is enabled prior to a write operation only. W is enabled once the write operation is complete as is maintained at a high voltage. Pre-charge signal PC, Data signal, Write Enable WE, and sense amplifier enable SAE signals find the same use as those in 6T SRAM cell.

6.2 8T SRAM Memory Cell:

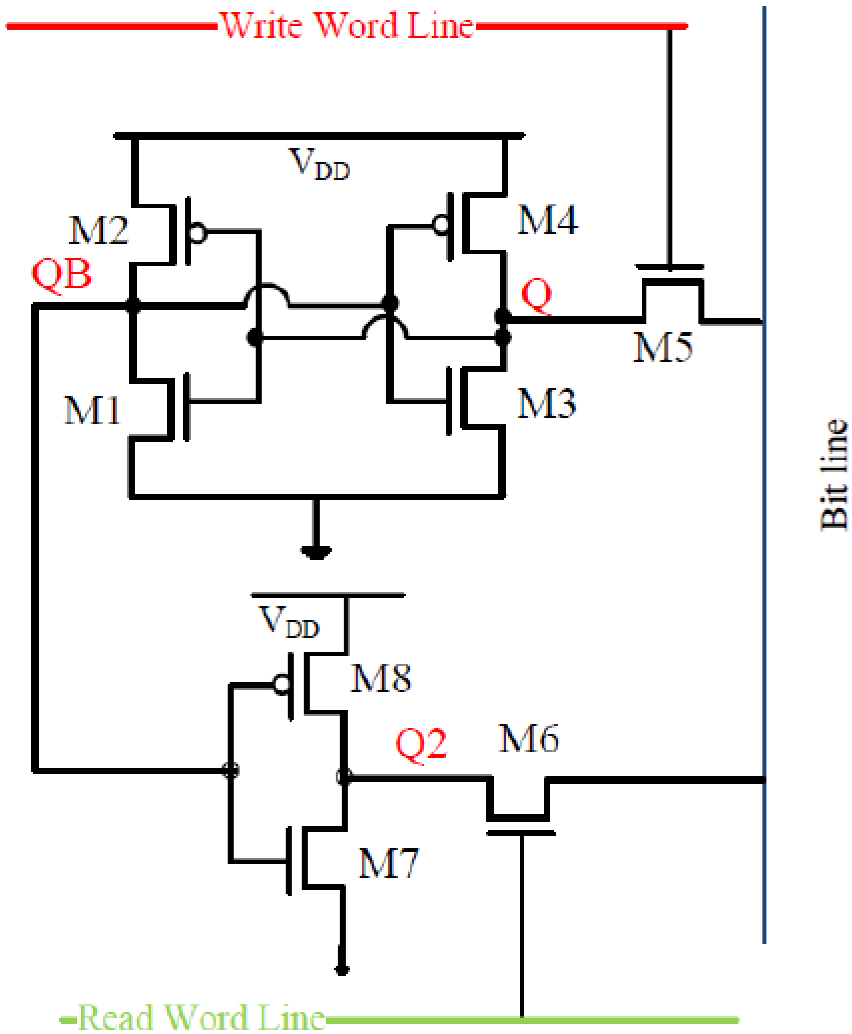


Figure15: The schematic for 8T SRAM cell

Figure shows the schematic representation of an 8T SRAM cell. The design employs a common bit line for both 'read' and 'write' operations, which results in less power consumption. The 8T SRAM is known to have a high read static noise margin, reason being, there are three extra transistors M6, M7 and M8 isolating the read path from the write path [14].

6.2.1 Write:

Like in the design of a 7T SRAM, a single bit line is used. BL is asserted to the value to be written, read word line remains turned off, while the write word line is asserted and M5 is enabled. Data nodes Q and QB are complimentary to each other so that, if logic 1 has to be written into the cell, Q starts charging, which enables the transistor M1 and a 0 is stored on the data node QB. This is followed by turning on M4 to facilitate the storage of a good logic 1 into Q.

6.2.2 Hold:

Hold cycle involves concealing the bit line from the three inverters. In order to decrease the hold SNM for this cell, data stored in QB drives the two inverters.

6.2.3 Read:

Read path comprises of the transistors M6, M7 and M8. On asserting the read word line, the transistor M6 is enabled so that the data is ready to be read from Q2.

The graphs obtained from the simulation of the 8T SRAM cell are shown in the figure below. It shows the output waveforms at BL, PC, RWL, WWL, Q, Q2 and QB.

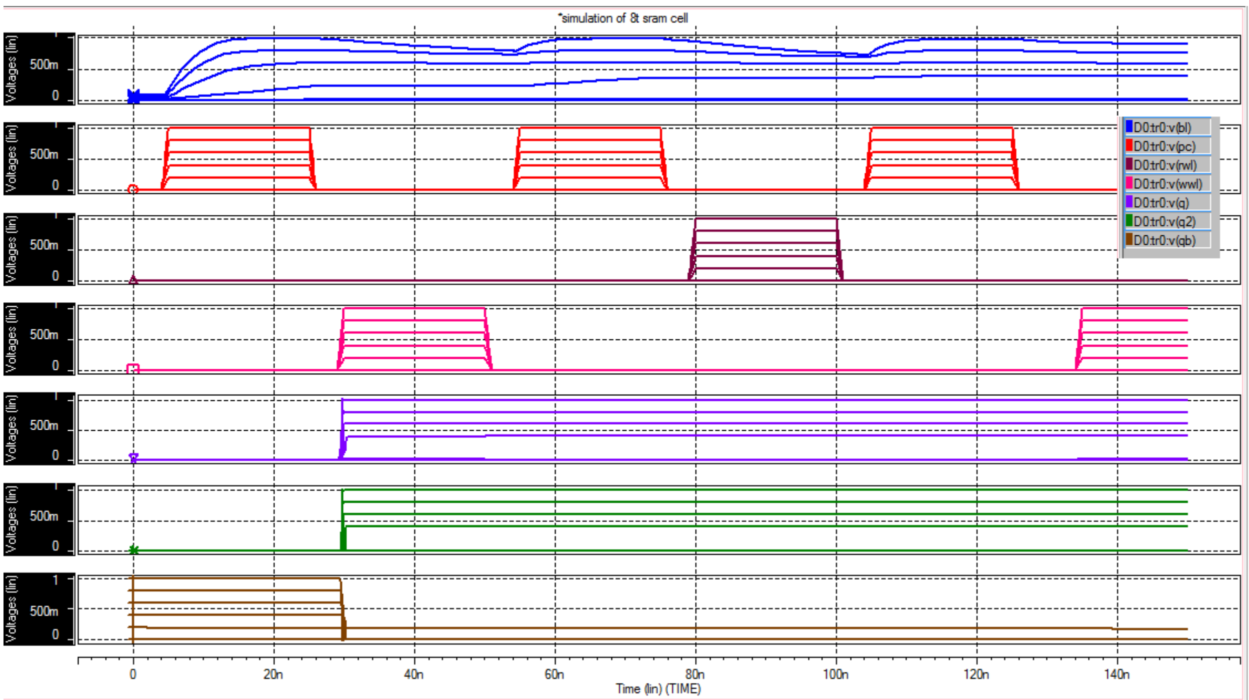


Figure16: Simulation of 8T SRAM cell

Pre-charge signal PC, Data signal, Write Enable WE, and sense amplifier enable SAE signals find the same use as those in 6T SRAM cell. 8T SRAM cell uses separate input signals for real and write operations (RWL and WWL) because it has a single bit line for both read and write mechanisms. An additional reference signal is required to start the sensing process, as there is a single bit-line from memory.

6.3 9T SRAM Memory Cell:

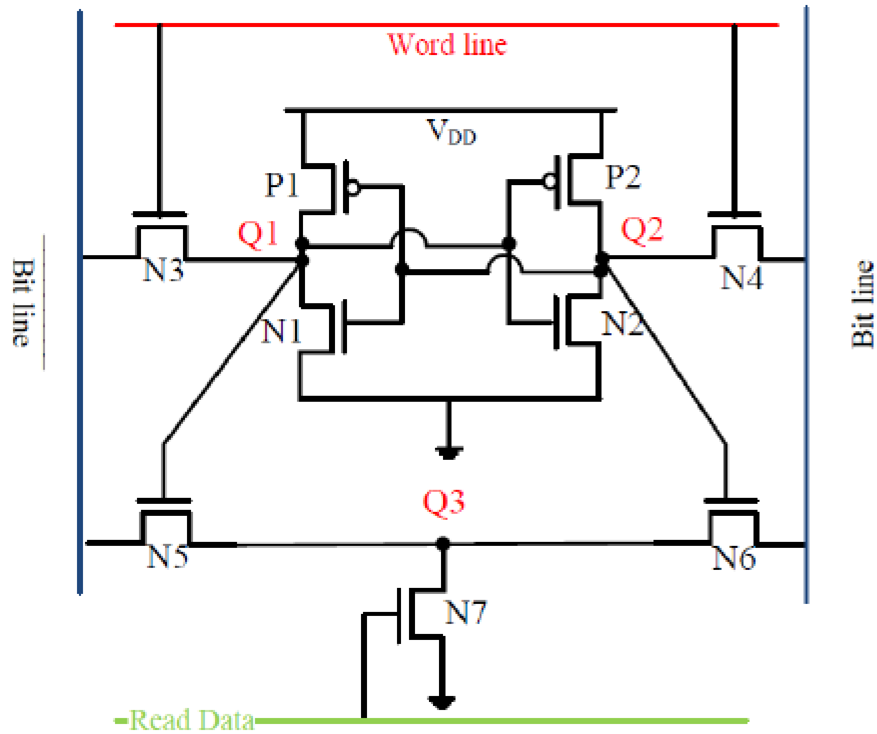


Figure17: Schematic representation of a 9T SRAM cell

Both read and write SNM's are very high in a 9T SRAM cell design. The read and write operations employ two different data access mechanisms. The data nodes are isolated from the bit lines during a read cycle, which results in an elevated read SNM as compared to the conventional 6T SRAM memory cell. The schematic of a 9T SRAM memory cell is shown in the figure. The word line regulates the access gates N3 and N5. Nodes Q1 and Q2 store the incoming data. The data stored in the data nodes in turn controls the access transistors N5 and N6. Read data line guides the transistor N7 [15].

6.3.1 Write:

The value to be written, and its compliment are fed to the bit lines BL and BL' respectively. Transistors N3 and N4 are enabled by asserting the word line, maintaining the read data line low. Nodes N3 or N4 receive the input data to be written.

6.3.2 Hold:

The hold state is relatively less stable because a part of the data node will still be connected to the bit lines while the word line and read line are turned off.

6.3.3 Read:

When the read cycle begins, the read word line is turned on while the write word line is kept off. Suppose a logic 1 is stored at data node Q2, the transistor M6 is turned on, allowing current flow from

BL’ to ground through the transistors M6 and M7. The sense amplifier detects the difference in voltage between BL and BL, which is then followed by the data read operation.

The graphs obtained from the simulation of the 9T SRAM cell are shown in the figure below. It shows the output waveforms at BL, BL’, WR, Data, WE and RD.

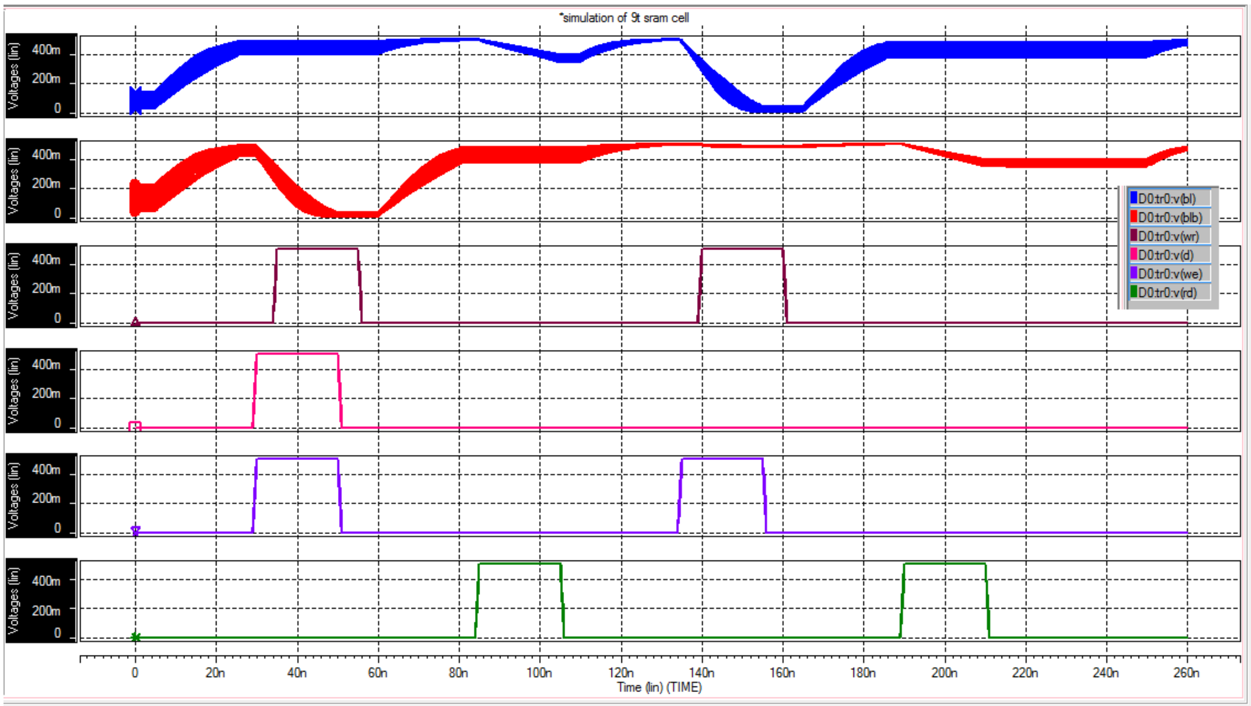


Figure18: Simulation of 9T SRAM cell

Pre-charge signal PC, Data signal, Write Enable WE, and sense amplifier enable SAE signals find the same use as those in 6T SRAM cell. The pre-charge signal has to be cautiously enabled since the memory is never physically disconnected from the bit-lines. During a write operation, WR signal is enabled and RD signal is used to initiate a read operation. The read operation differs from that of other memory cells.

6.4 10T SRAM Memory Cell:

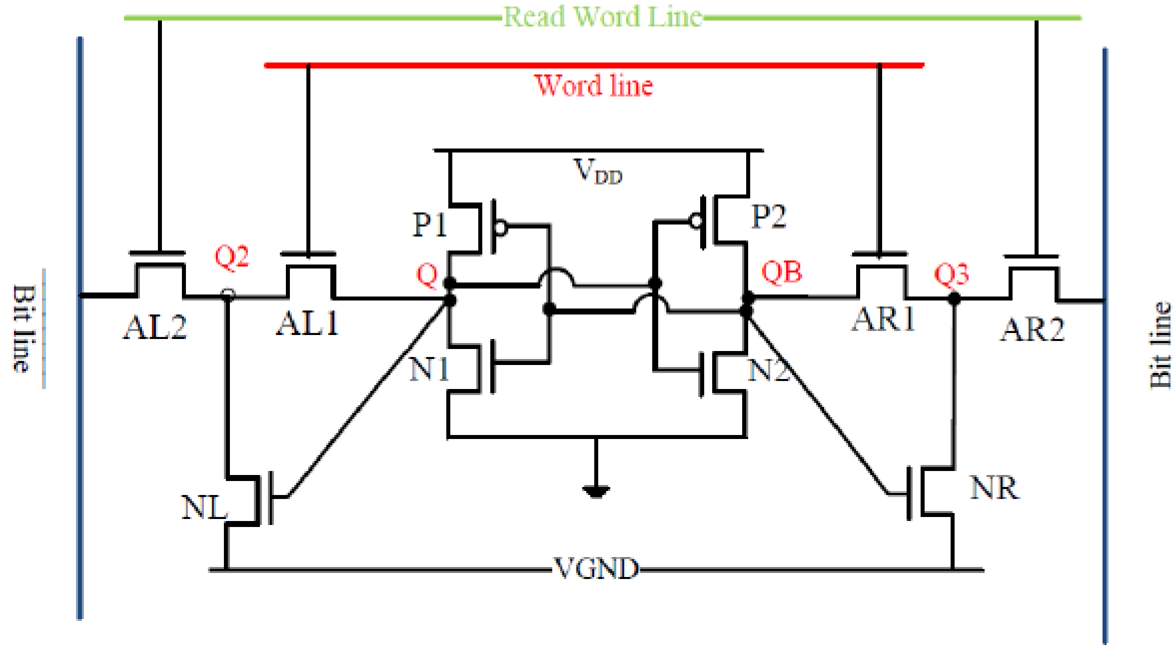


Figure19: Schematic representation of a 10T SRAM cell

This SRAM cell design offers better read stability. Adequate bit inter-leaving is granted for immunity against soft errors. Figure shows the schematic of this memory cell. The read and write operations differ slightly from the 9T SRAM design though the structure of the two is quite similar [16].

6.4.1 Write:

Both read and write word lines are enabled and VGND (Virtual Ground) is maintained high. Bit lines BL and BL' need to be at a higher potential than VDD because of the weak writability resulting from the two extra access gates in this cell design. Data nodes Q and QB hold the input data and its complement.

6.4.2 Hold:

Keeping the virtual ground high both read and write word lines are disabled and the cell is isolated from the bit lines.

6.4.3 Read:

Virtual ground is maintained at the ground potential, read word line is turned off and the write word line is enabled. Disabling the word line results in the isolation of the data nodes Q and QB from the bit lines, this improves the read SNM.

The graphs obtained from the simulation of the 10T SRAM cell are shown in the figure below. It shows the output waveforms at BL, BL', WL, WWL, and VGND.

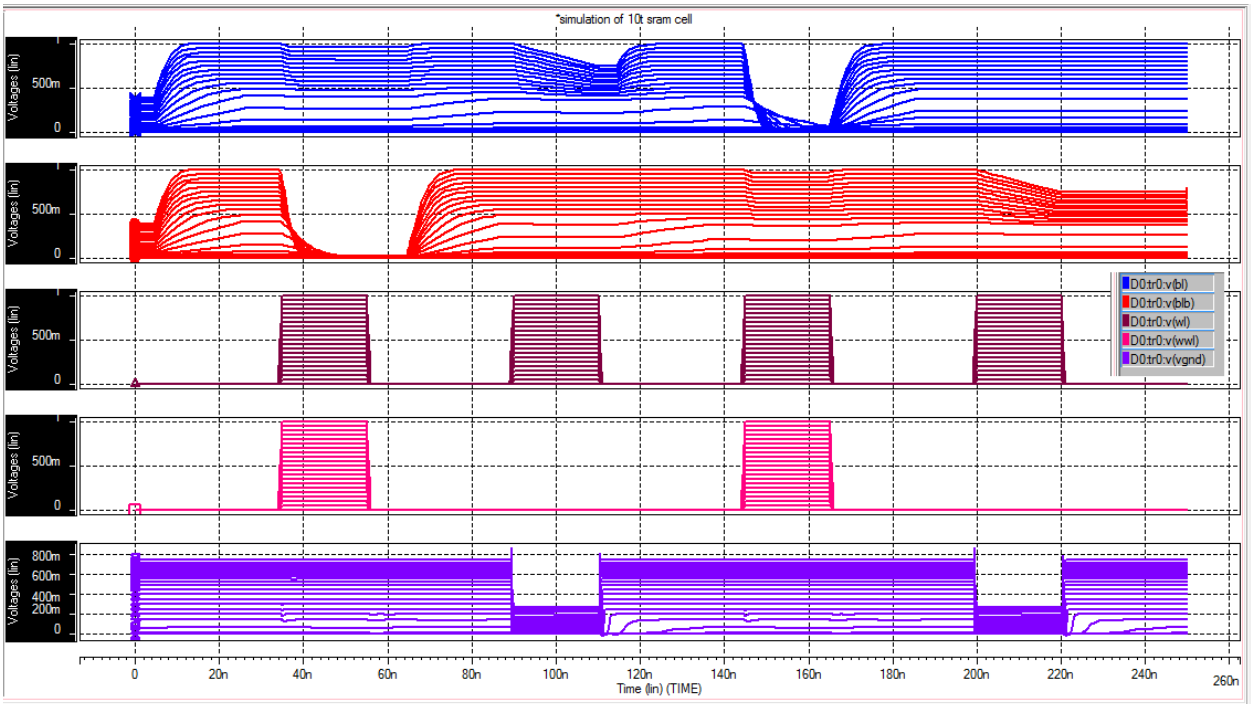


Figure20: Simulation of 7T SRAM cell

The read operation in the 10T SRAM cell uses a virtual ground technique. Clock signal (CLK) along with the read cycle enabled signal, generate the virtual ground control signal. The read and write mechanisms are similar to those in the 9T SRAM cell. Pre-charge signal PC, Data signal, Write Enable WE, and sense amplifier enable SAE signals find the same use as those in 6T SRAM cell. WL signal is used to trigger a read cycle. During a write operation, the signals WL and WWL are set.

7. PERFORMANCE EVALUATION AND COMPARISION OF DIFFERENT SRAM CELLS

Several custom designed SRAM cells were simulated, evaluated and compared against other possible designs. All simulations and design were performed using HSPICE with 22, 32 and 45nm Berkeley Predictive Technology Model to study their power consumption, access time variation, with the variation in supply voltage, metal oxide layer thickness, and threshold voltage. Threshold voltage and Oxide thickness variations are realized using the Gaussian Parameter distribution AGAUSS function in HSPICE. All the leakage values are calculated under 4 different values of temperature, -40^{o}C, 27 ^{o}C, 100 ^{o}C and 125 ^{o}C.

7.1 Stability of SRAM cells:

The cell stability is an important aspect of SRAM cell design, which arbitrates the sensitivity of the memory to external conditions and process variations and determines the soft error rate. Static Noise Margin (SNM) is the essential parameter for studying the stability of SRAM cells under process variation and supply voltage reduction. Transistor mismatch and offsets resulting from manufacture variations and variation in operating conditions, give rise to a DC disturbance termed as static noise. The challenge is to design an SRAM cell, which under all operating conditions, has sufficient noise margin to combat disturbances caused by cross talk, thermal noise, supply voltage noise and alpha particle disturbances. SNM value is estimated by means of a circuit simulator. Detailed procedure is explained in the following section.

A DC noise voltage source (V_n) is placed in series with the cross-coupled inverters with the internal nodes of the cell being assigned with the worst case polarity. SNM is obtained by locating the largest square within the butterfly characteristic curve. Measurement of SNM for a 6T SRAM cell has been explained in detail in the section to follow.

7.2 Process Variations:

International Technology Roadmap for Semiconductors (ITRS) proposed the below table which shows that the variation on threshold voltage, increases along the feature scaling. The figures shown in the table are those of the designs known to have high performance.

Channel Length L(nm)	250	180	130	90	65	45
Threshold voltage VT (mV)	450	400	330	300	280	200
Variation on VT (mV)	21	23	27	28	30	32
(Variation on VT)/VT	4.70%	5.80%	8.20%	9.30%	10.70%	16%

Table3: Variation of threshold voltage with the increase along the feature scaling

Studying the butterfly curves obtained by the voltage transfer characteristics and inverse voltage transfer characteristics (VTC and VTC-1) gives us a good understanding of the stability of a SRAM cell. The read static noise margin (SNM) of a standard 6T SRAM cell is shown in figure (a). Due to process variation, constriction of the SNM window can be observed. At lower VDD=0.3v, this effect is more pronounced as shown by the figure (a). Reliability and performance are hence strongly disturbed by process variations at lower supply voltages [1]. In order to bypass the read SNM problems in SRAM cells, various designs have been proposed of late, which provide many times better read SNM even under low supply voltages. Figures (b) and (b') show the butterfly curves of an 8T SRAM cell which is known to be read SNM free. [1]

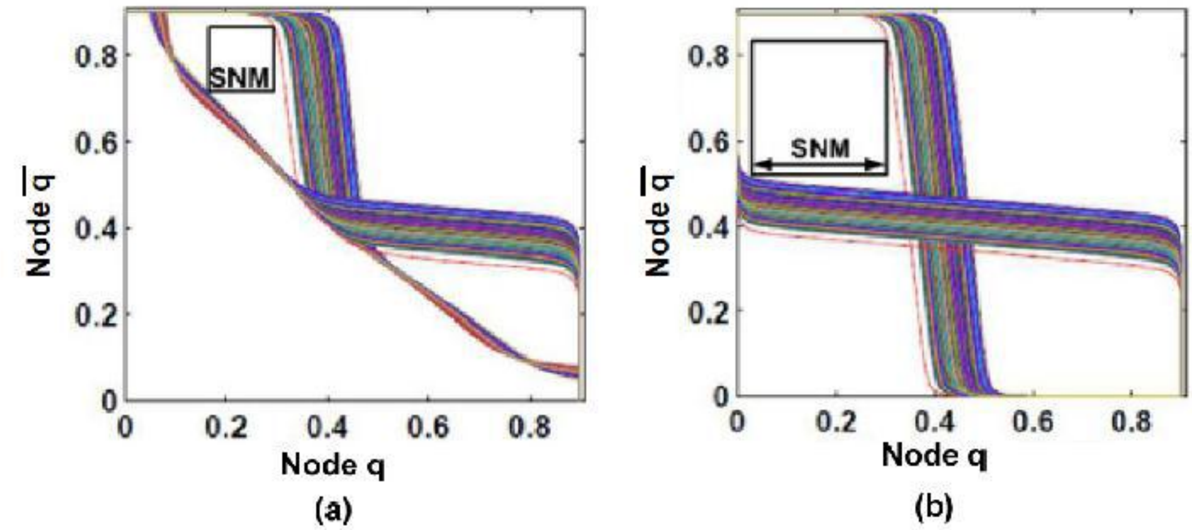


Figure21: Measurement of read static noise margin (SNM) at VDD=0.9V for 45nm technology node (a) standard 6T SRAM cell, and (b) read SNM free 8T SRAM cell. [1]

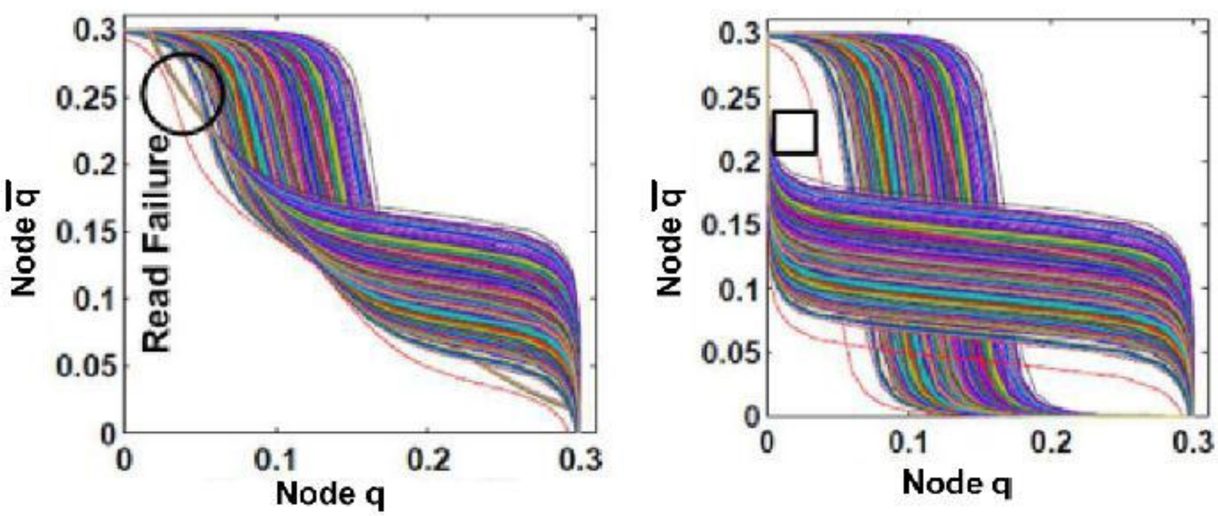


Figure22: Measurement of read static noise margin (SNM) at VDD=0.3V for 45nm technology node (a) standard 6T SRAM cell, and (b) read SNM free 8T SRAM cell. [1]

7.3 Static Noise Margin curves of SRAM cells:

7.3.1 Hold SNM:

Figure shows the equivalent circuit for hold state SNM measurement. The corresponding butterfly curves for each of the memory cell designed are attached below. Hold state is when the memory is not being accessed, but the data has to be retained. Hold margin determines the minimum supply voltage or delta retention voltage in standby mode. The static noise margin in hold state is generally used to evaluate the hold stability, the word line being set to zero voltage. A static noise V_n, the maximum value of which is V_{DD}, is introduced using a DC voltage source and DC analysis is performed for output voltage QB measurement and then the output Q is mirrored to it. Extensive simulations are performed using HSPICE, and the output binary file obtained is then compiled by the C compiler, after which MATLAB tool is used to plot the voltage transfer characteristic curve.

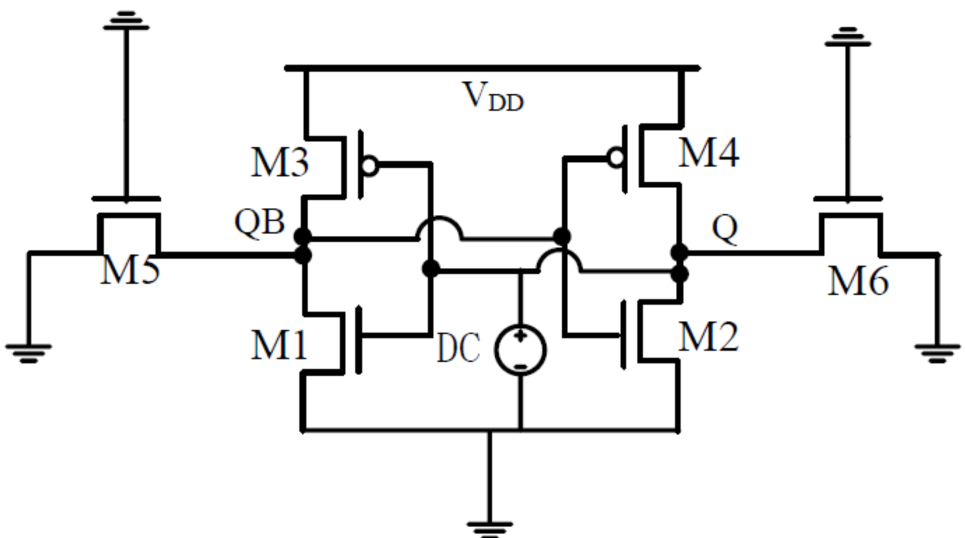


Figure23: Circuit for hold state SNM measurement [2]

Figure below shows the hold state static noise margins of all the proposed SRAM cell designs (6T through 10T). It can be observed that as the number of transistors within the SRAM cell increases, the SNM for hold state goes on improving and 10T SRAM cell has the best SNM in hold state. Still, we cannot conclude that 10T SRAM cell is the most stable circuit among all others because, the SNM that has been measured, is at minimum operating voltage for every cell.

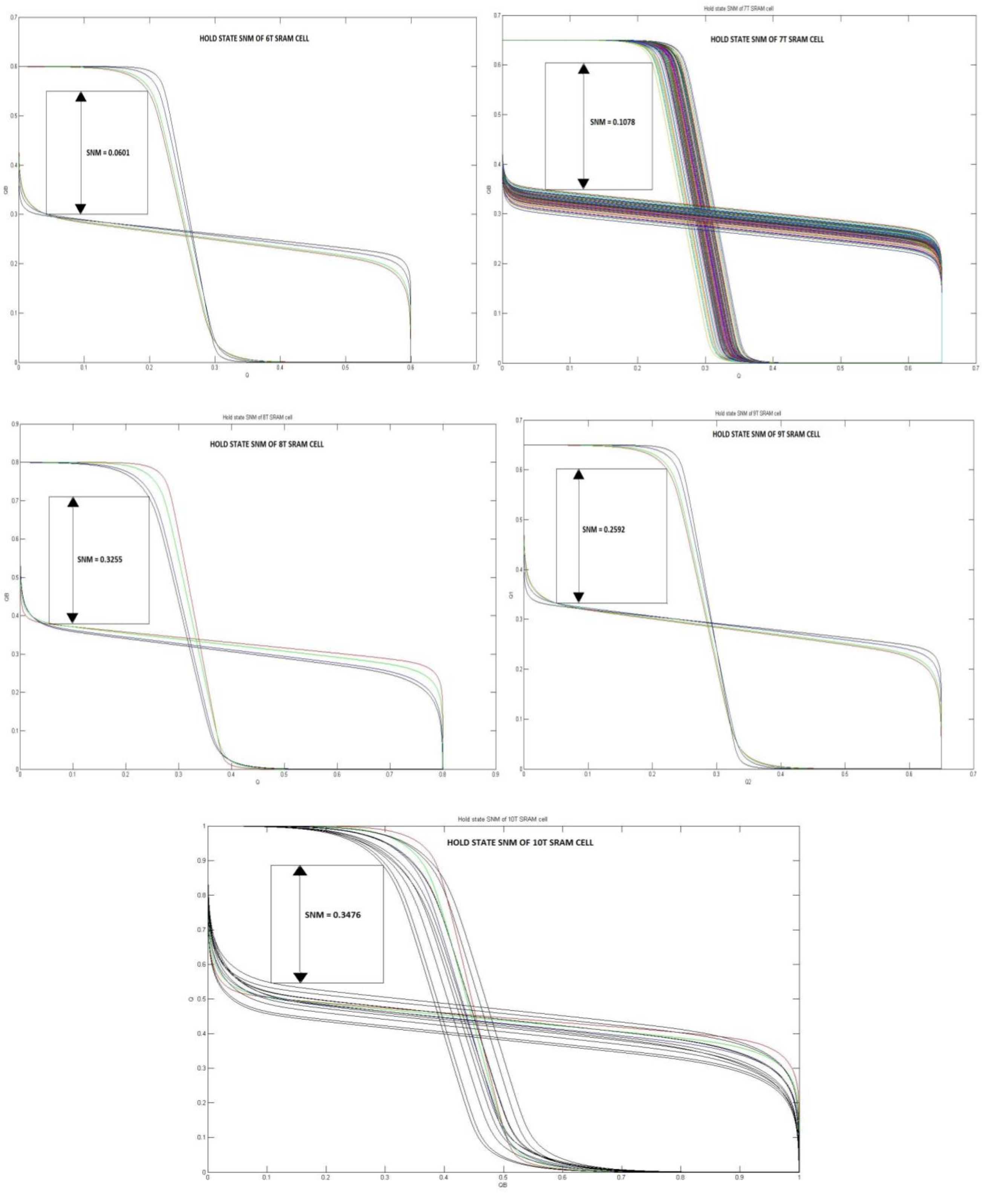


Figure24: Hold state SNM curves of 6T to 10T SRAM cells

Figure below shows the hold state SNM variation of a 6T SRAM cell for variation in supply voltage.

7.3.2 Read SNM:

The circuit used for measuring the read state SNM of a 6T SRAM cell is shown in the figure below. If a logic ‘0’ which has been written into the memory cell has to be stored in the data node Q, the voltage at the node Q will be higher than zero during the read cycle, and this may change the data stored at node QB. Cell ratio is a parameter given by the ratio of pull-down transistors and access transistors. Cell ratio determines the read state static noise margin. In the circuit for calculating the read SNM, the DC voltage source V_n sweeps from 0 to V_{DD} in the read mode. This voltage is given to the logic ‘0’ data node and the relevant voltage transfer curve is drawn.

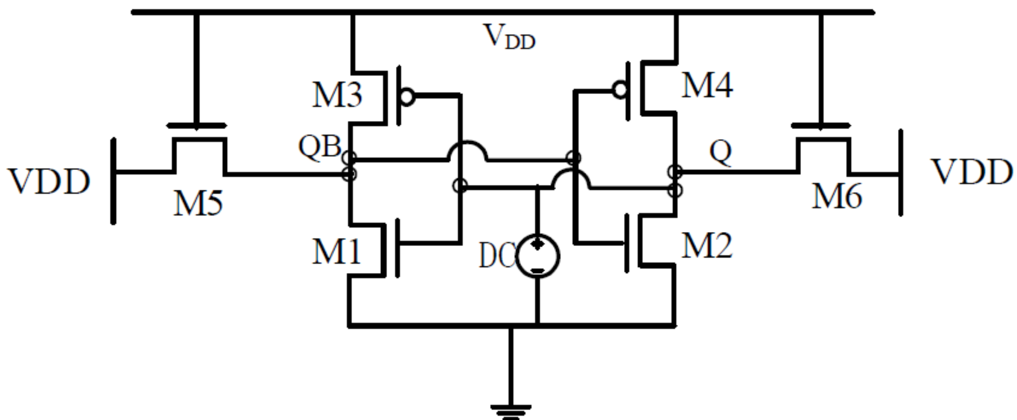
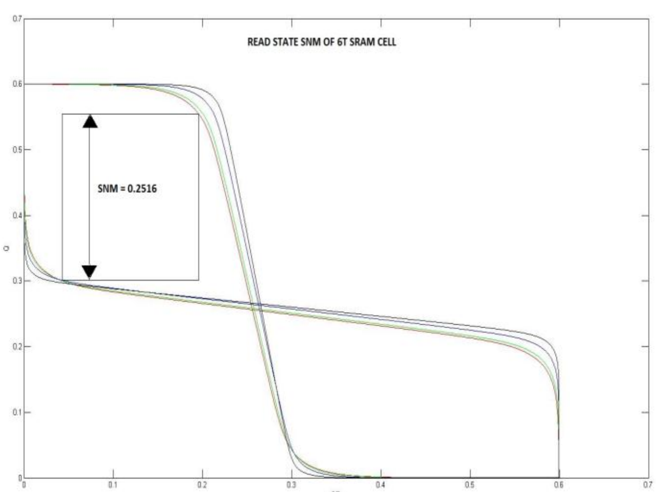
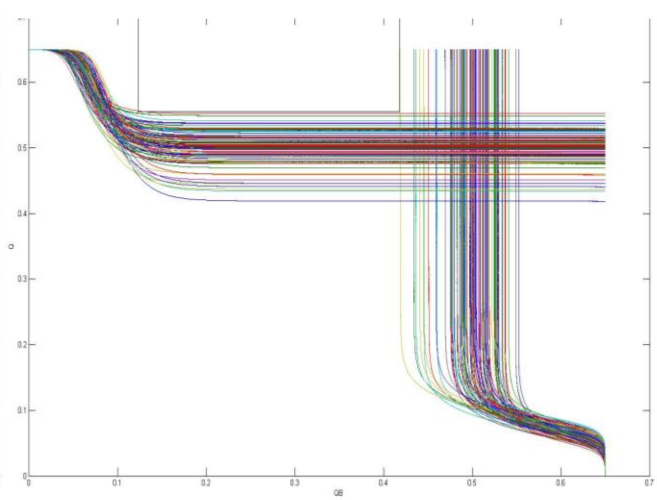


Figure25: Circuit used for measuring the read state SNM of 6T SRAM cell [2]

The read state SNM curves simulated for 6T-10T designs is shown below.





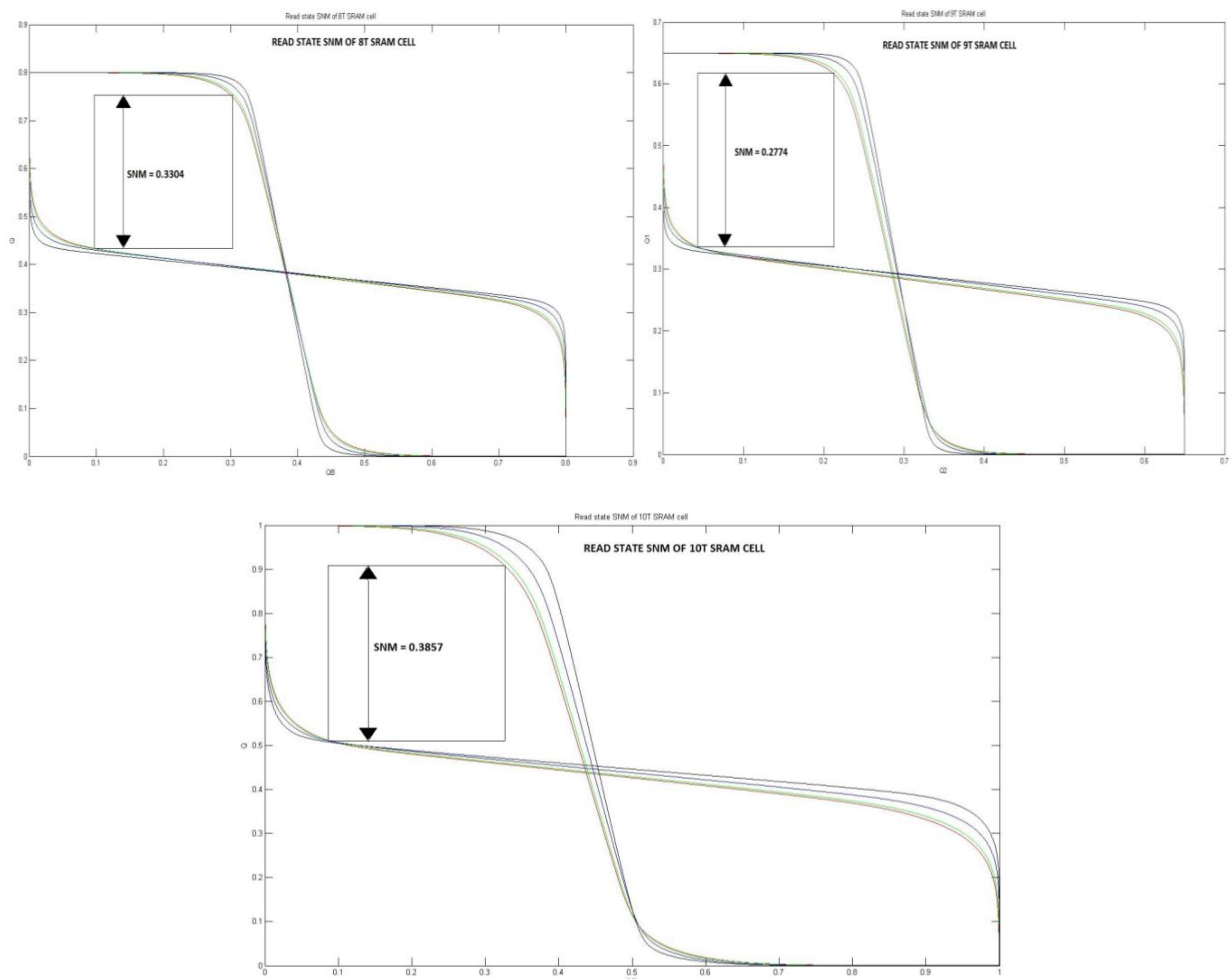


Figure26: Read state SNM curves of 6T to 10T SRAM cells

As observed from the figures, it is clear that 7T SRAM cell has the smallest read static noise margin. The reason being, there is an additional transistor in the feedback path between the two cross coupled inverters, because of which, the pull-down network is burdened with extra load. 9T and 10T SRAM cells have good read stability, owing to the isolated data node technique employed.

7.3.3 Write SNM:

Circuit built to calculate the write state SNM of 6T SRAM cell is shown in the figure below. The butterfly VTC curves obtained for different SRAM cells are also attached. The maximum voltage on the bit-line needed to flip the cell content is termed as the '*write trip voltage*'. Write state SNM depends largely on this write trip voltage. It is determined by the pull up ratio of the memory cell. To reduce power consumption, the bit-lines are generally pre-charged to a certain voltage level less than the supply voltage V_{DD}. The DC voltage source in the circuit setup to compute the write SNM is used to simulate the write trip voltage. .

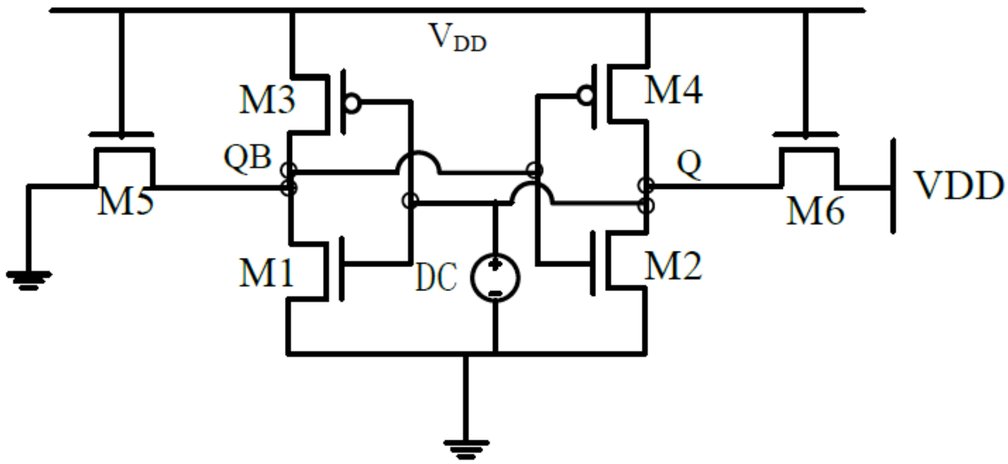
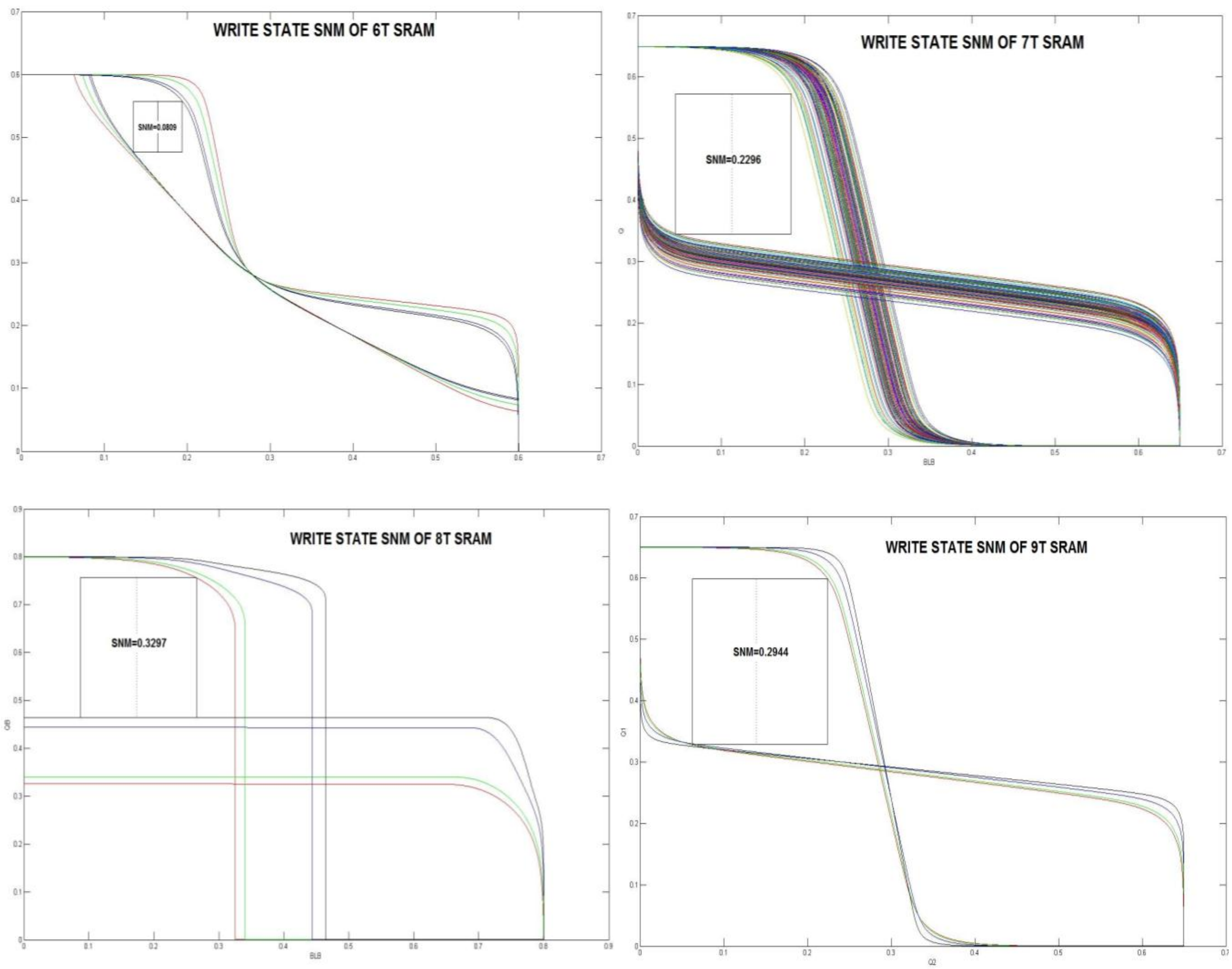


Figure27: Circuit used for measuring the read state SNM of 6T SRAM cell [2]



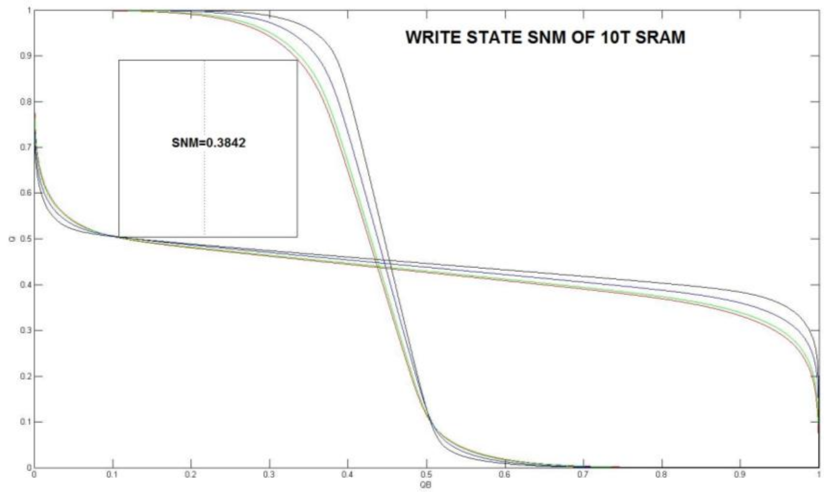


Figure28: Read state SNM curves of 6T to 10T SRAM cells

The write mechanism of the 8T SRAM cell slightly differs from that of the other cells. It uses only one bit line in order to reduce the write trip voltage. In case of 7T SRAM cell, the feedback path is cut off between the two inverters, when the write cycle is just about to start. Write trip voltage has a major influence on power consumption during the write cycle. Hence, it is advantageous to keep the write trip voltage to its minimum value.

The numerical results are tabulated in the table below. The supply voltage was kept constant at 1V during these simulations.

SRAM Cell	Hold State SNM	Read State SNM	Write State SNM
6T	0.0601v	0.2516v	0.0809v
7T	0.1078v	0.1502v	0.2296v
8T	0.3255v	0.3304v	0.3297v
9T	0.2592v	0.2774v	0.2944v
10T	0.3476v	0.3857v	0.3842v

Table4: Static Noise Margins of various SRAM cell designs

The table helps us to infer that 8T and 10T SRAM cells are more stable and give better performance for the same supply voltage. Isolation of the cell nodes from their bit-lines helps a great deal to improve the read stability. Write stability is improved by disconnecting the feedback path between the two cross coupled inverters. Although cell complexity, cell area and power consumption for these designs is more compared to the standard 6T SRAM cell, the trade-off can be chosen as per the design requirement.

7.4 SNM variation with the supply voltage scale-down:

Stability of the SRAM cells is studied by varying the supply voltage. Lowering the supply voltage is one of the best techniques that can be employed to reduce energy consumption for ultra-low power operations. However, the performance of an SRAM array is highest when the supply voltage is at its fullest value. As the supply voltage reduces, the cells tend to lose their stability and data retention properties. For our analysis, V_{DD} was varied from 0.4V to 1V. Figure shows the variation in hold state static noise margin with varying supply voltage V_{DD}. It is clear that the SRAM cell is more stable at higher values of supply voltage. Graphs below show the variation of the hold state static noise margin of the SRAM cells designed by varying V_{DD}. The 9T SRAM cell performs data retention only when the supply voltage is at its maximum value that is 1V.

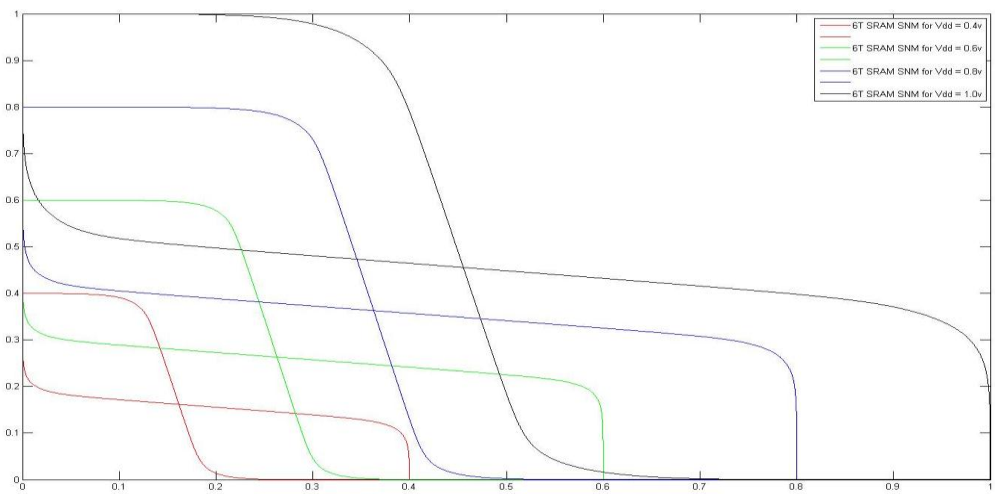


Figure29: Hold SNM variation of a 6T SRAM cell, with variation in V_{DD}.

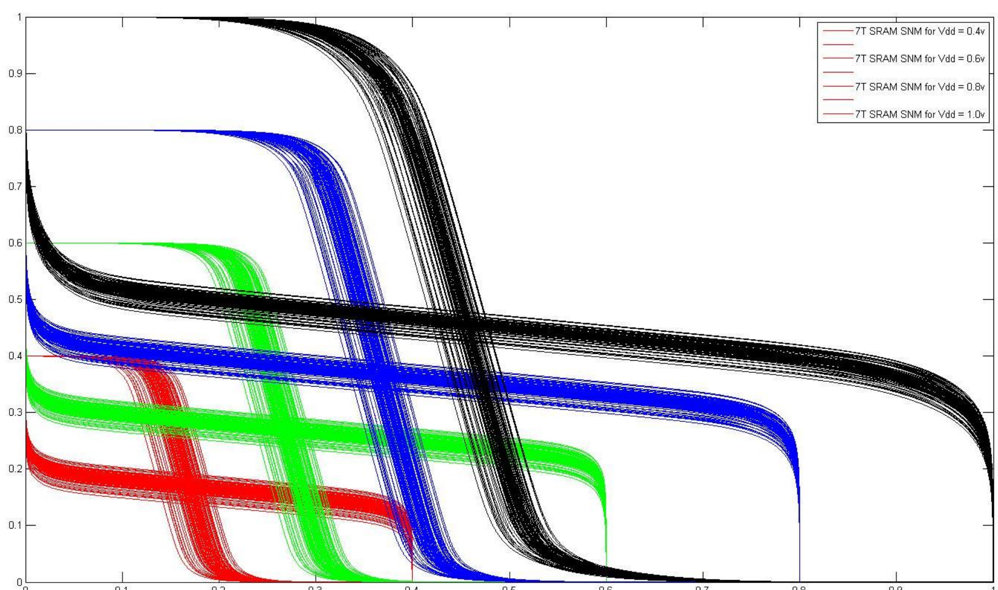


Figure30: Hold SNM variation of a 7T SRAM cell, with variation in V_{DD}.

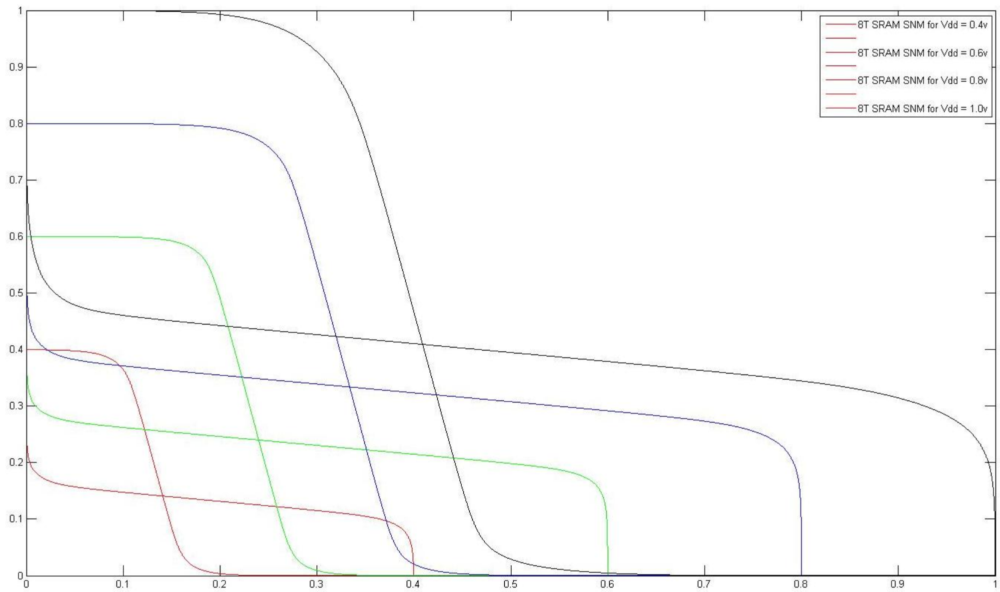


Figure31: Hold SNM variation of an 8T SRAM cell, with variation in V_{DD}.

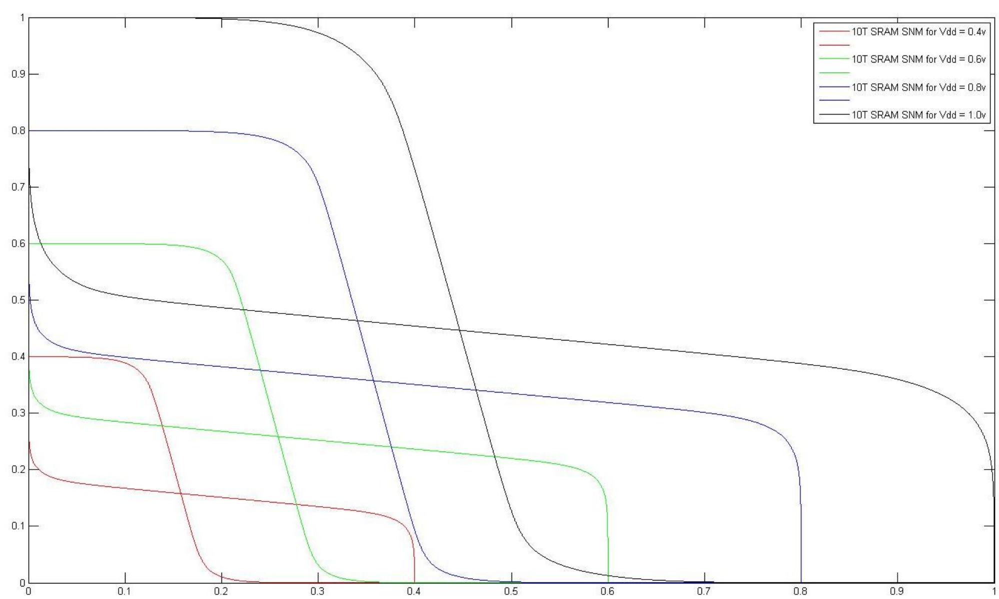


Figure32: Hold SNM variation of a 10T SRAM cell, with variation in V_{DD}.

8. <u>EFFECT OF PROCESS VARIATION</u>

Read stability, write stability, hold stability, access time, and power consumption are the parameters used for evaluating the performance of the SRAM cells under the effect of process variation.

8.1 Effect on read and write operations:

Write time is the delay between the first rise of the WL signal to 50% of the supply voltage and the rise or fall of voltage at the data node Q to 50% of V_{DD}. *Read Time* is the delay between the second rise of WL to 50% of V_{DD} and fall of voltage at one of the bit lines to 10% of V_{DD}. Time consumed for writing data into a memory cell and reading it back is referred to as *Access Time.* It is important to decide on the minimum operating supply voltage for each of the memory cells as Access Time depends greatly on the supply voltage. Extensive Monte Carlo Simulations were performed in order to study the effect of process variation on memory cells. Graphs below show the failure rate of different SRAM cells. These were obtained by taking 1000 iterations by varying the oxide thickness and threshold voltage using the Gaussian distribution function in HSPICE.

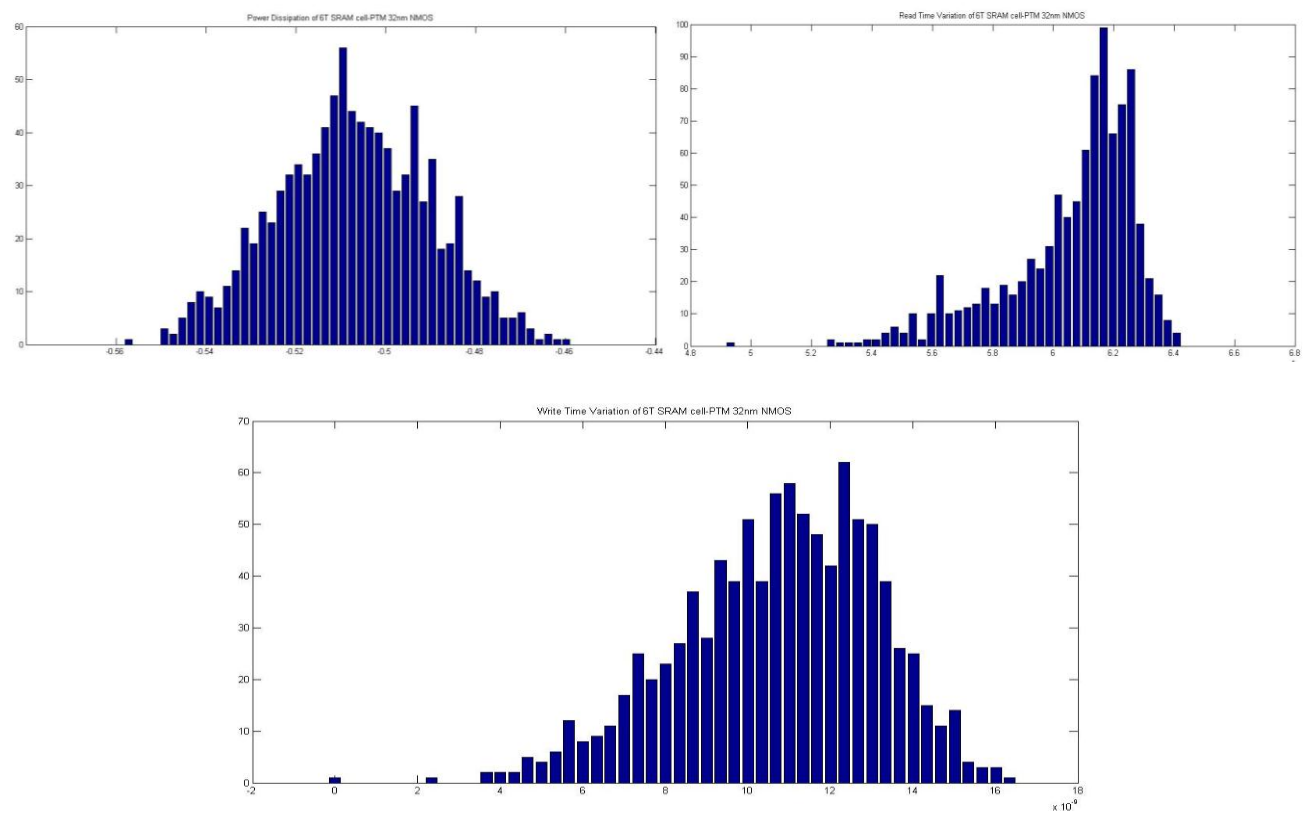


Figure33: Power dissipation, read time, write time variation of 6T SRAM cell

The figures clearly conclude that the read operation in a 6T SRAM cell has a higher failure rate than the write operation at a supply voltage as low as 0.5 volts. Dynamic power is also high because, one of the two bit-lines has to be discharged during every read and write operation.

Figure below shows the failure rate for the 7 transistor SRAM memory cell. Similar Monte Carlo simulations were performed with 1000 iterations varying the oxide thickness and threshold voltage.

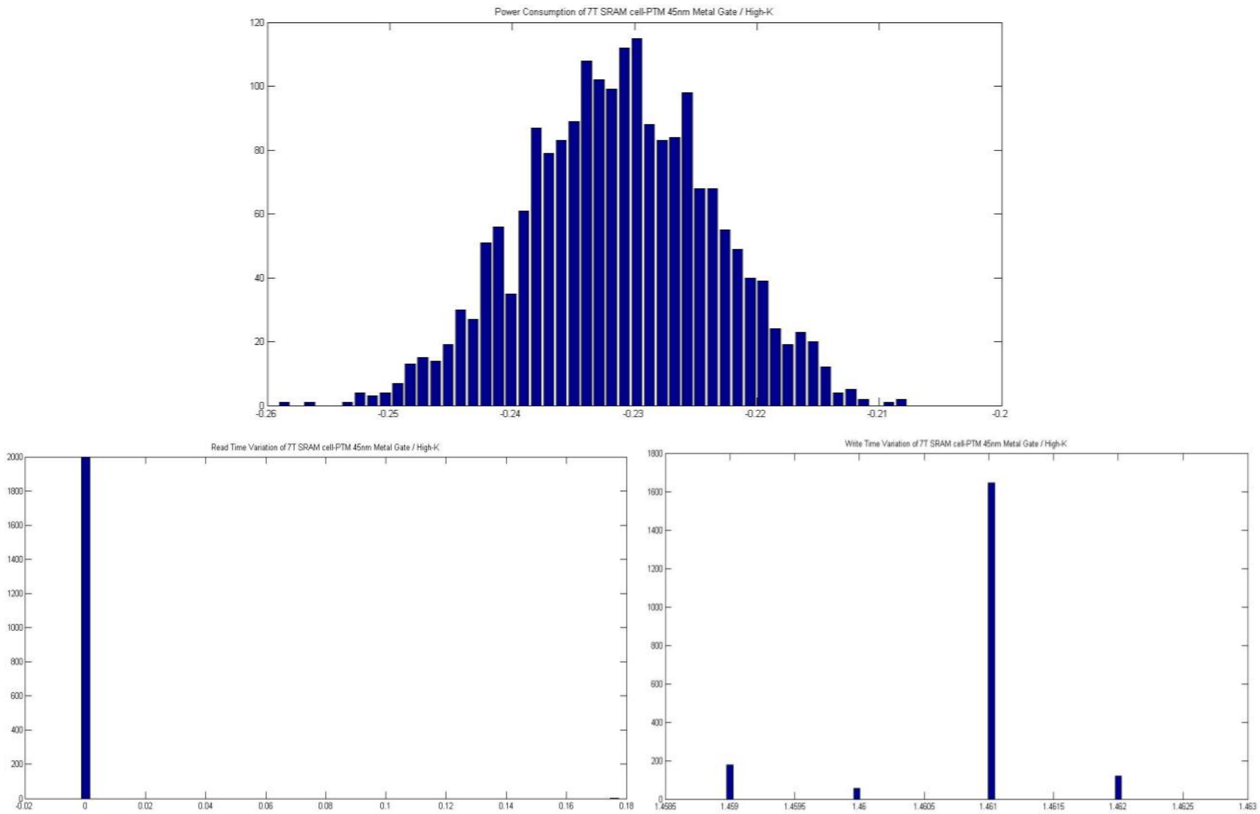


Figure34: Power dissipation, read time, write time variation of 7T SRAM cell

From the figures, it is evident that the read and write mechanism of 7T SRAM is very reliable even at lower supply voltages. Write operation is significantly accurate and the write time is reasonable under the effect of process variation.

Failure rate of 8T SRAM cell is shown in the figures below. The simulation techniques are similar to those used for 6T and 7T SRAM cells.

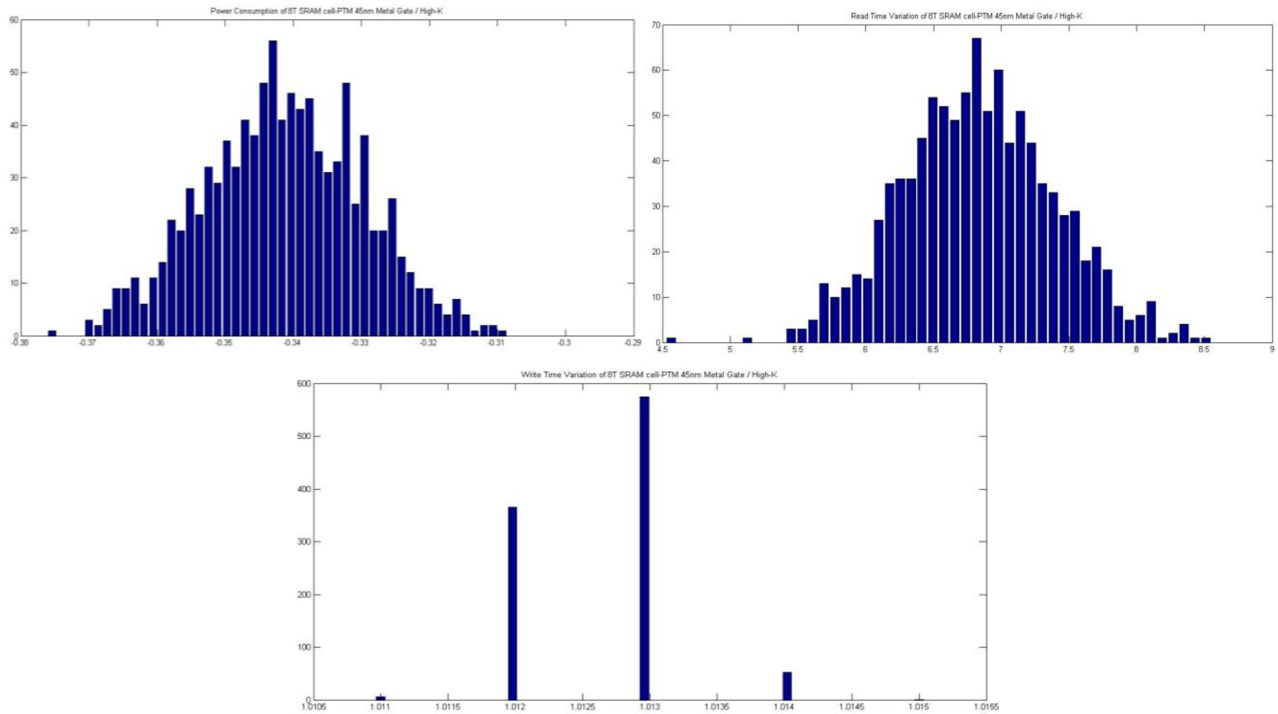
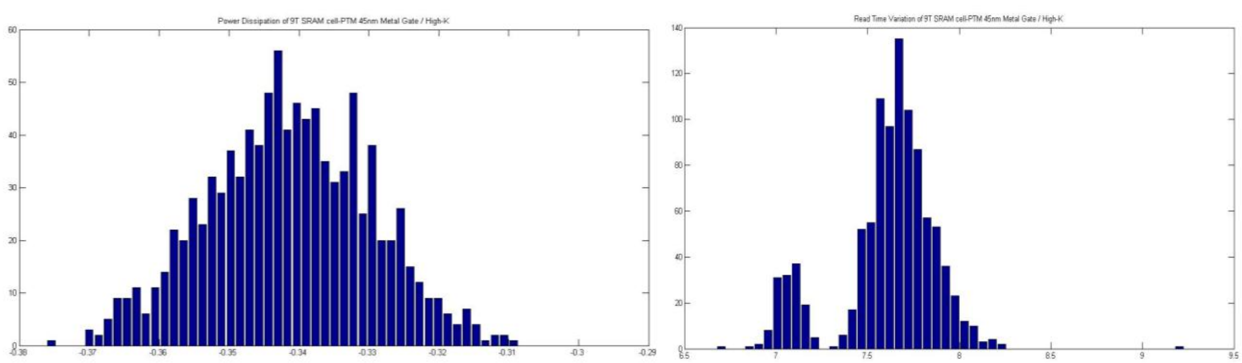


Figure35: Power dissipation, read time, write time variation of 8T SRAM cell

Single bit line is used for write operation in the proposed 8T SRAM cell. From the figure, it can be seen that the write operation is quite reliable although there is some vulnerability in the read operation. Read operation requires the data node to be separated using a single inverter. For accurate write operation, the write access transistor has to be very efficient.

Failure rate of 9T SRAM cell for read time, write time and power consumption parameters are shown in the figures below.



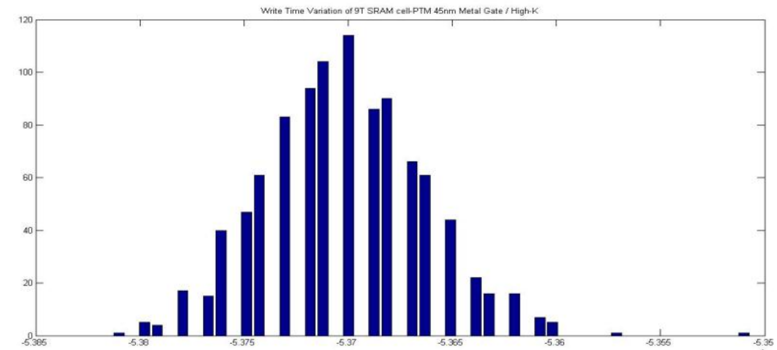


Figure36: Power dissipation, read time, write time variation of 9T SRAM cell

Write operation of the proposed 9T SRAM cell is very much similar to that of the standard 6T SRAM cell. Read operation consumes more power as it uses two extra transistors in order to keep the data node separated from the bit lines. The figures show that the read operation failure is greater at lower voltage levels. The write operation is considerably stable although variations are present for lower voltages.

Failure rate of 10T SRAM cell is shown by the figures below for similar Monte Carlo operations.

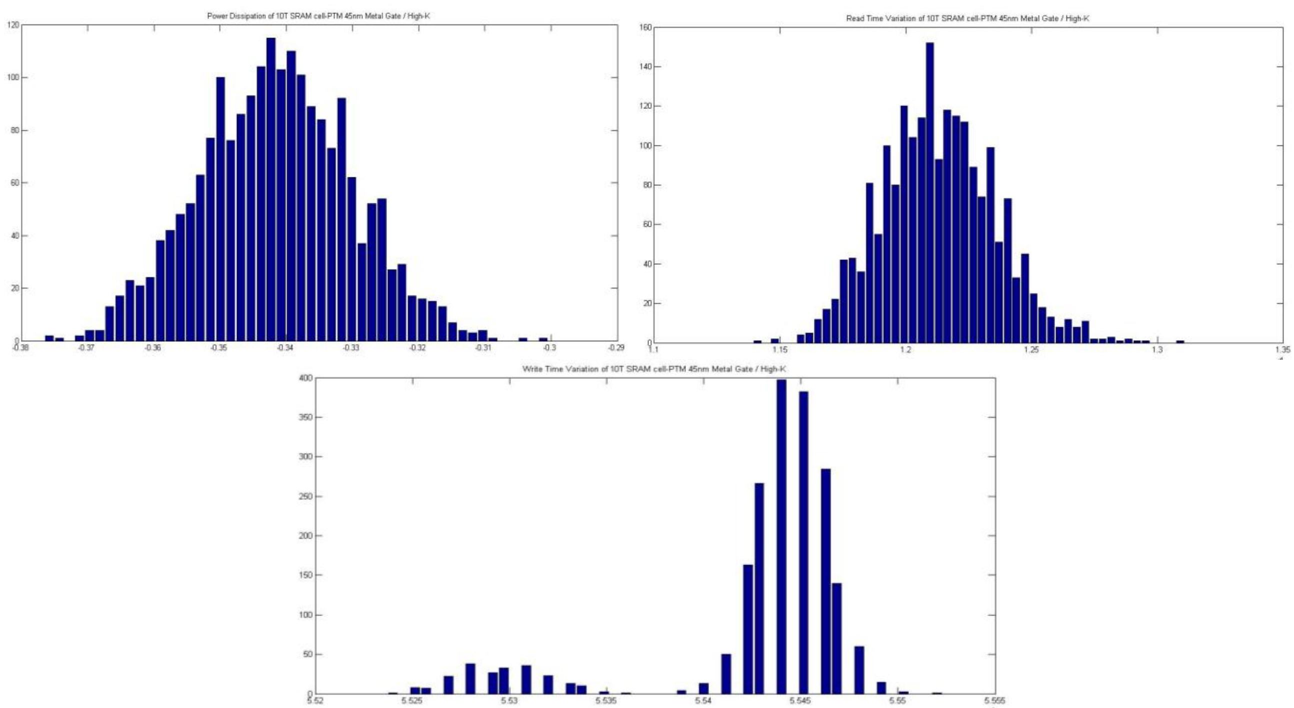


Figure37: Power dissipation, read time, write time variation of 10T SRAM cell

The read and write operations of the 10T SRAM cell are very similar to that of the 9T SRAM cell as explained above. It is evident that the write operation of the 10T SRAM cell is more reliable than that of the 9T SRAM cell.

8.2 Numerical analysis:

Numerical analysis of the read time, write time and power consumption parameters for the proposed memory cell designs are explained in detail in the section below. Mean and Standard deviation of each of the above mentioned parameters have been computed to be able to evaluate and compare the performance of the cells better.

Table below shows the mean and standard deviations for the read time variation of all the memory cells, and the figure following shows the Matlab simulations for read time variation for all the five designs of memory cells.

45nm CMOS	READ TIME	VDD=0.5v
	MEAN	STANDARD DEVIATION
6T	5.27E-09	1.46E-10
7T	1.46E-07	3.52E-11
8T	1.05E-07	5.26E-10
9T	5.14E-08	4.30E-09
10T	5.14E-08	1.46E-08

Table5: Mean and standard deviations for the read time variation of all the memory cells

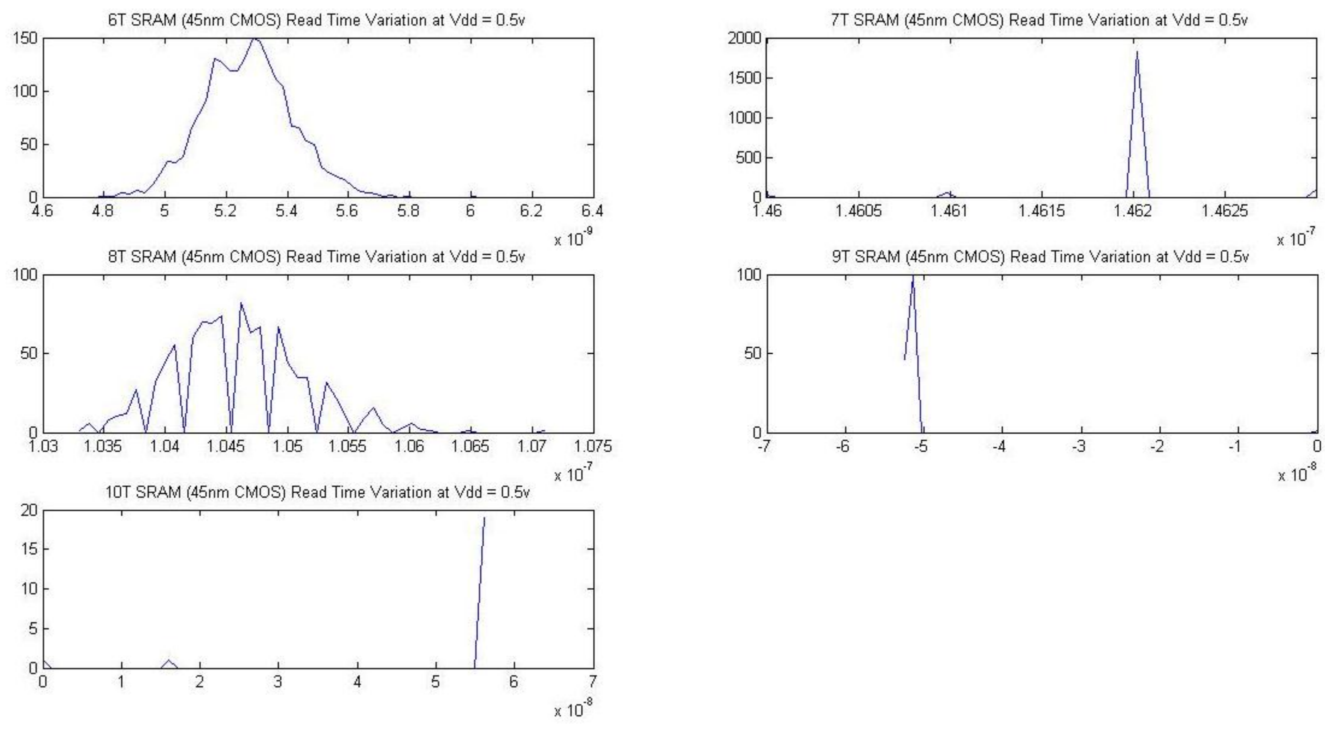


Figure38: Read time variation on 6T-10T SRAM cells

From the results of simulation and the graphs, we can conclude that apart from the standard 6T SRAM cell, the read time of 9 and 10T SRAM cells are minimum and stable when compared to other cells. Although the read time of 8T SRAM cell is also good, it has a large variation as compared to other cells.

Mean and standard deviations for the write time variation of all the memory cells are tabulated as shown by the table below, and the figure that follows shows the write time variation as obtained by matlab simulations. It can be seen that 7T and 9T SRAM cells have best write times without much variations and the write time mean values for these cells are also minimum.

45nm CMOS	**WRITE TIME**	**VDD=0.5v**
	MEAN	**STANDARD DEVIATION**
6T	**8.52E-12**	**1.68E-11**
7T	**4.44E-08**	**3.67E-12**
8T	**5.02E-10**	**5.55E-11**
9T	**1.12E-07**	**9.37E-09**
10T	**5.37E-09**	**1.41E-09**

Table6: Mean and standard deviations for the write time variation of all memory cells

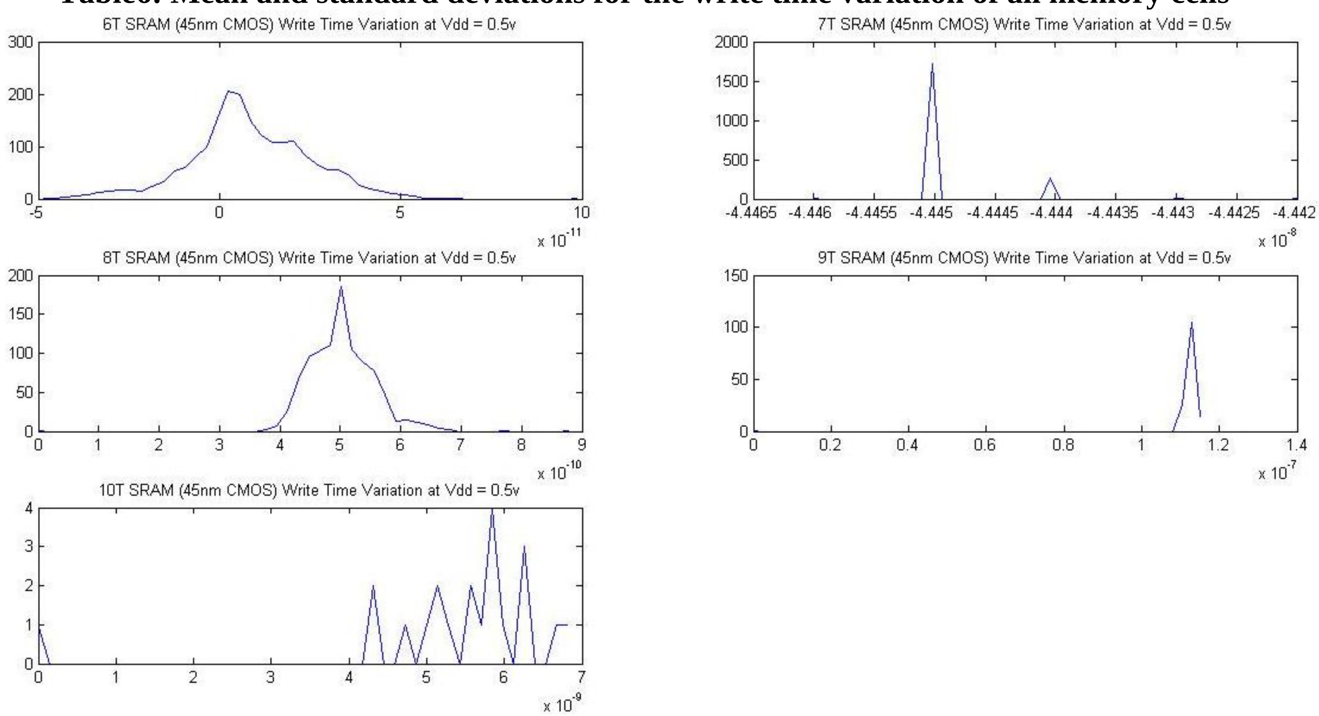


Figure39: Read time variation on 6T-10T SRAM cells

Table below shows the mean and standard deviation values computed for the power consumed by the 5 SRAM cell designs. It can be seen that 7T SRAM cell consumes least power when compared to all other memory cells, and the variation is also reasonable.

45nm CMOS	**POWER CONSUMPTION**	**VDD=0.5v**
	MEAN	**STANDARD DEVIATION**
6T	**3.42E-01**	**1.14E-02**
7T	**2.31E-01**	**7.72E-03**
8T	**3.42E-01**	**1.15E-02**
9T	**3.41E-01**	**1.13E-02**
10T	**3.43E-01**	**1.03E-02**

Table7: Mean and standard deviations for the power consumption of all the memory cells

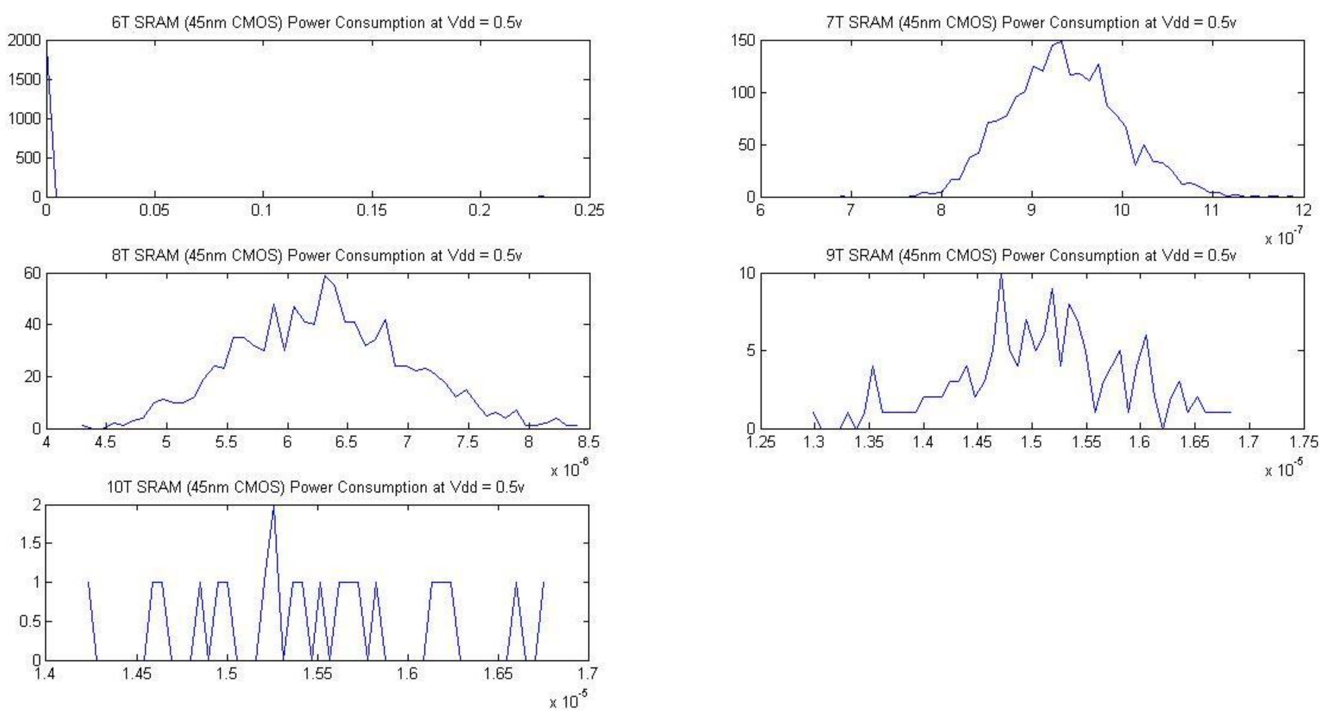


Figure40: Read time variation on 6T-10T SRAM cells

9. <u>ERROR CORRECTION CODING: (ALGORITHM LEVEL)</u>

9.1 HAMMING CODE:

Among all the error correction codes that exist, Hamming code has been the most popular and widely used code, especially in the area of SRAM design, because of its accuracy and simplicity. It was invented by Richard Hamming in 1950.

Hamming code is most useful for detection and correction of one bit error in an encoded word. The approach is very appropriate because the occurrence of single bit errors is more likely than change in two or more bits. A hamming code word is comprised of data bits and parity bits. Hamming distance is defined as the number of bits that differ between two words. Minimum hamming distance is the least bit difference between all possible word pairs in a set of words. A hamming code with a minimum distance d_{min} equal to 3 and the length of the parity bits j greater than or equal to 3 can be defined using the following parameters:

Error correction capability: $t = [(dmin-1)/2] = 1$
Data-word length: k
Parity bits length: $j = [\log 2k] + 1$
Code-word length: $n = k+j$

For instance if there are 4 data bits and 3 check bits, then the total length of the encoded word is 7 bits and this has a single bit error correction capability.

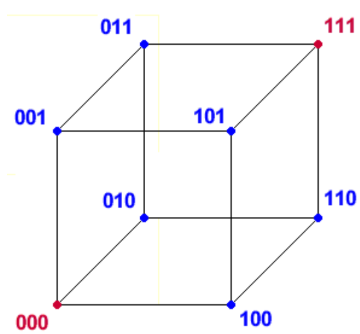


Figure41: Hamming Cube

The design mechanism can be classified into three steps:

I) For every n data bits, k bits of parity are inserted so that the following equation is satisfied:

$$\mathbf{2^k \geq n + k + 1}$$

Parity bits are inserted at positions 1, 2, 4….$2^{(k-1)}$, to elongate the original data word. The remaining positions are occupied by the actual data bits.

II) Parity bits are determined using modulo 2 addition of selected bits in the (n+k) word

$\mathbf{P_x \oplus B_{x,1} \oplus B_{x,2} \oplus ... = 0}$

Where P_x stands for the Parity Bit, and the data bits that are to be selected are represented by B_x's. The data bits on which modulo 2 addition is to be performed are selected based on the position of the parity bit P_x. The following method is applied to determine the data bits to be XORed.

For P1 (parity bit at the first position): skip 0 bits, mark 1 bit, skip 1 bit, mark 1 bit;

For P2 (parity bit at the second position): skip 1 bit, mark 2 bits, skip 2 bits, mark 2 bits;

For P4 (parity bit at the fourth position): skip 3 bits, mark 4 bits, skip 4 bits, mark 4 bits; [27]

In general, for a Parity bit at the xth position, skip (x-1) bits, mark x bits, skip x bits, mark x bits;

III) Decoding an encoded hamming word also uses the parity check equation and the process explained in step II.

$\mathbf{A_0 = P_1 \oplus B_{1,1} \oplus B_{1,2} \oplus ...}$

$\mathbf{A_1 = P_2 \oplus B_{2,1} \oplus B_{2,2} \oplus ...}$

$\mathbf{A_2 = P_4 \oplus B_{4,1} \oplus B_{4,2} \oplus ...}$

$\mathbf{A_{log2(x)} = P_x \oplus B_{x,1} \oplus B_{x,2} \oplus ...}$

The address word $A_{log2(x)}$ = A3A2A1A0 identifies the location of the error bit.

Let us consider an example to illustrate the use of hamming code to encode a 4 bit data. [28]

I) $\mathbf{2^k \geq 4 + k + 1 = 5 + k}$

Hence, k = 3. Parity bits occupy positions 1, 2 and 4 in the encoded word. So, the remaining bit positions 3, 5, 6 and 7 are occupied by the original data bits. So, the sequence of bits in the encoded word is P1P2D3P4D5D6D7.

II) Parity check equation for P_1 is

$\mathbf{P_1 \oplus B_3 \oplus B_5 \oplus B_7 = 0}$

Supposing the original data is D3D5D6D7 = 1010,

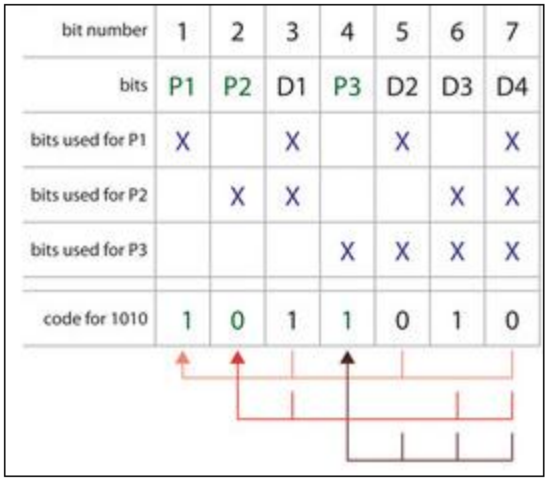
Then, $P_1 = 0$

Similarly, parity equation for P_2 and P_3 are

$\mathbf{P_2 \oplus B_3 \oplus B_6 \oplus B_7 = 0;}$

$\mathbf{P_3 \oplus B_5 \oplus B_6 \oplus B_7 = 0;}$

For the data bits 1010, $P_2 = 0$ and $P_3 = 1$ as illustrated by the table below:



bit number	1	2	3	4	5	6	7
bits	P1	P2	D1	P3	D2	D3	D4
bits used for P1	X		X		X		X
bits used for P2		X	X			X	X
bits used for P3				X	X	X	X
code for 1010	1	0	1	1	0	1	0

Figure42: Parity bits generation

Thus, the new encoded 7 bit word is 1011010.

III) Assuming that a single bit error occurs in the encoded word, and one of the bits is flipped, 00"0"1010, the decoder determines the location of the error as follows:

A0 = P1 ⊕ B3 ⊕ B5 ⊕ B7 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1 (Parity check failed)

A1 = P2 ⊕ B3 ⊕ B6 ⊕ B7 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 1 (Parity check failed)

A2 = P3 ⊕ B5 ⊕ B6 ⊕ B7 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0 (Parity check passed)

A2A1A0 = 011 = 3, pointing to the 3rd bit as the location of the error.

H = [P : I]

$$H = \begin{array}{c} \begin{array}{ccccccc} D_3 & D_2 & D_1 & D_0 & P_2 & P_1 & P_0 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Figures below show the simulation waveforms of (7,4) hamming code circuit

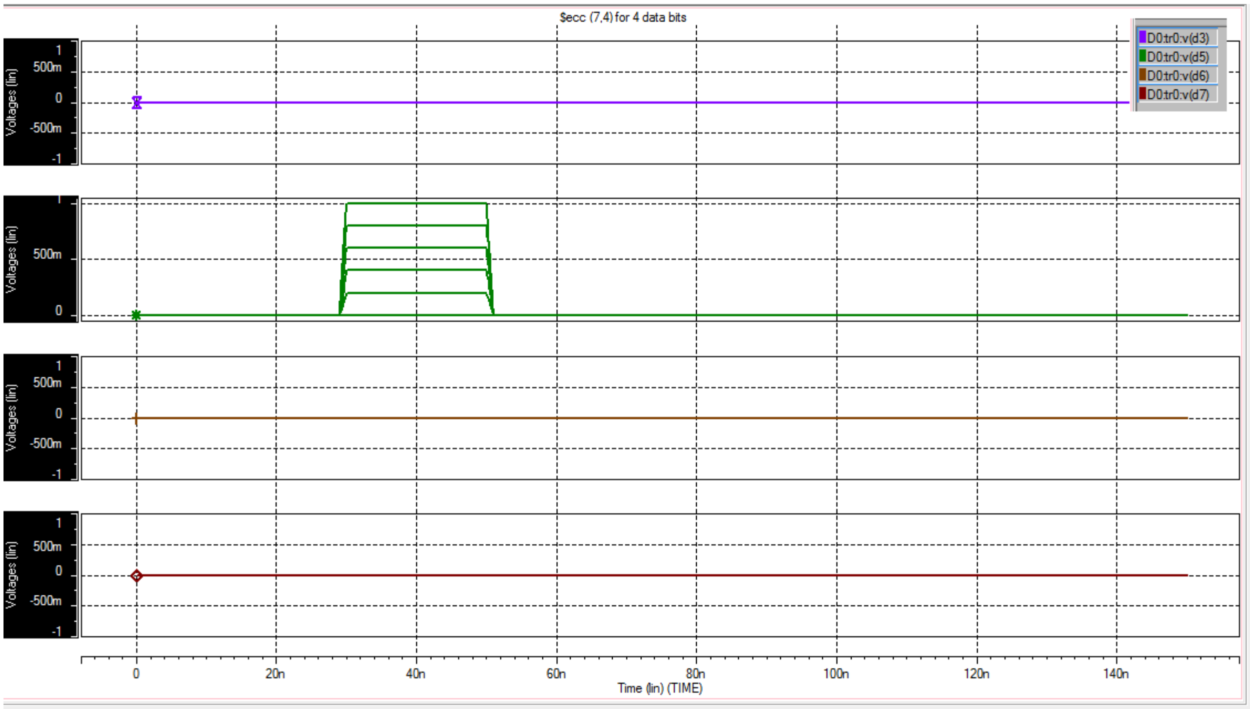


Figure43: Data bits (0100)

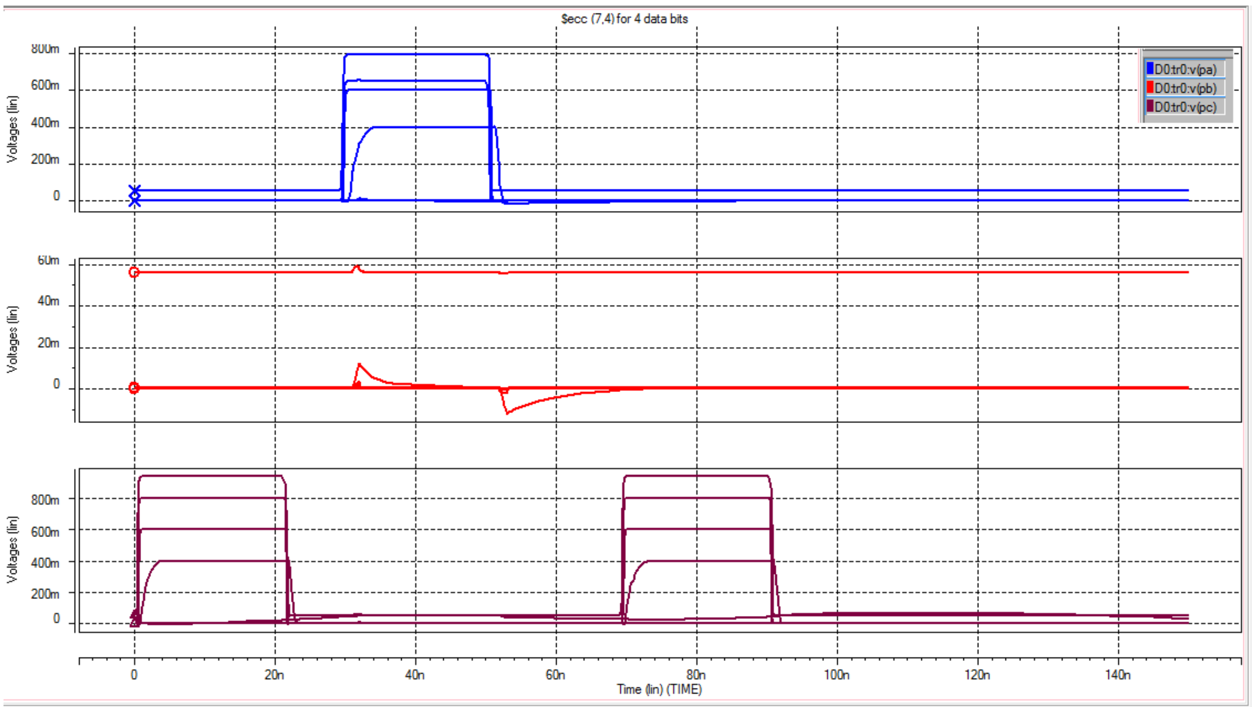


Figure44: Parity Bits generated by modulo 2 additions (101)

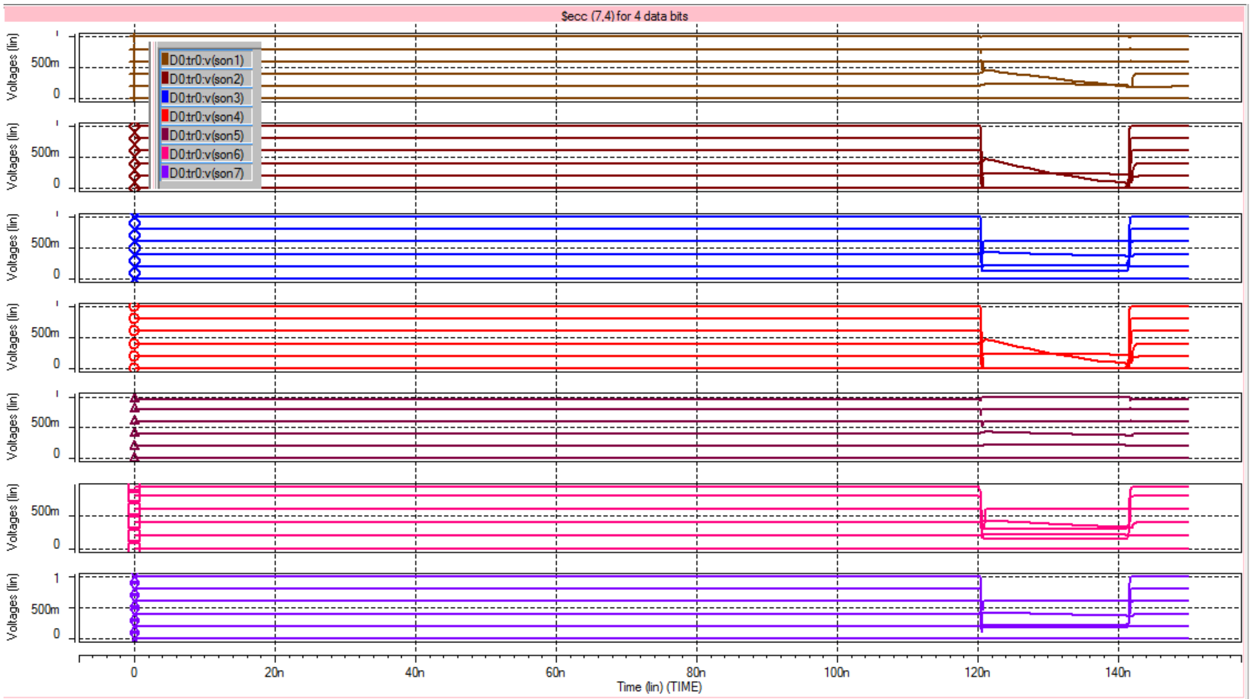


Figure45: Data read from the array of SRAMs

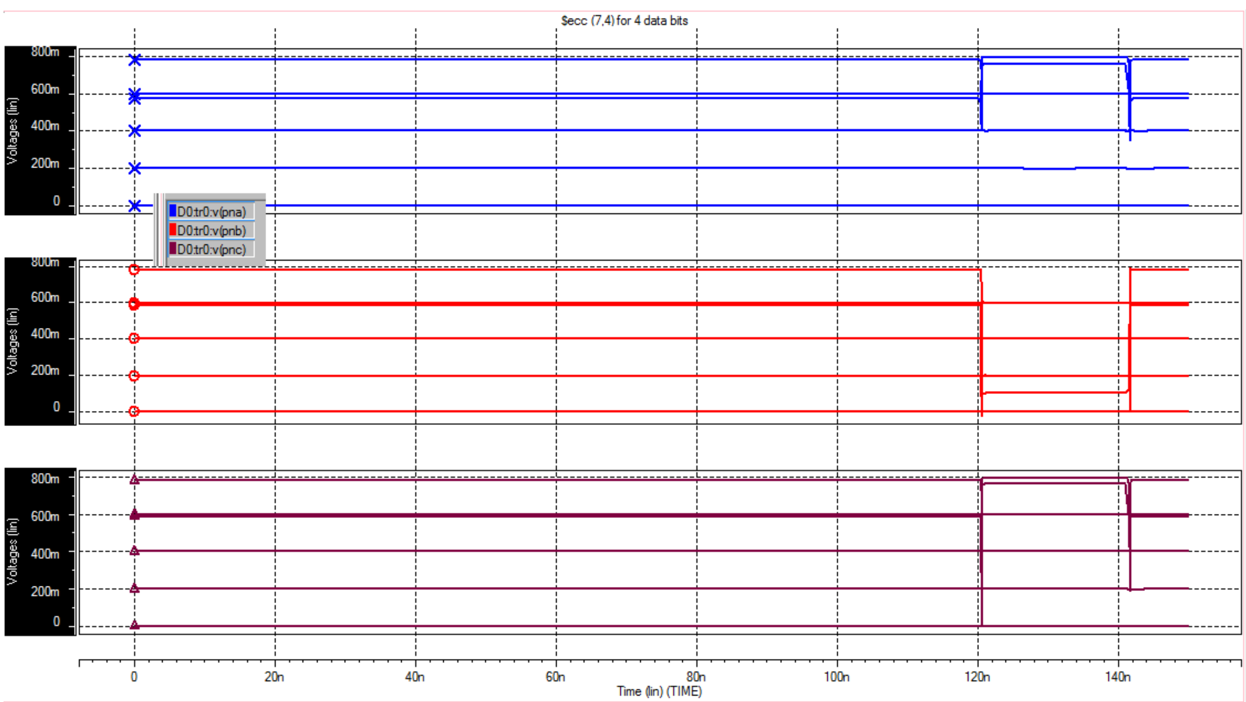


Figure46: Syndrome bits

Table below shows the different combinations of hamming weight circuits simulated and the time each one of them took for error detection, write operation and syndrome bits generation.

	No. of SRAM cells required (seconds)	Write Time (seconds)	Syndrome time delay (seconds)	Time for error detection (seconds)
7,4,3	**7**	**17.5602n**	**627.3036p**	**10.54444n**
8,4,4	**8**	**17.5553n**	**629.6965p**	**10.54444n**
11,8,3	**11**	**17.5271n**	**1.4894n**	**11.1261n**
12,8,4	**12**	**17.5296n**	**1.4947n**	**11.1213n**

Table8: Numerical Analysis of different Hamming codes

9.2 MODIFIED HAMMING SEC-DED CODES:

Hamming code with minimum distance $d_{min} = 4$ can detect double bit errors along with correcting single bit errors (Single error correction and double error detection). This can be achieved by extending a distance-3 hamming code with an extra parity bit that checks on all the symbols. The code thus obtained is known as modified hamming SEC-DED code. Multiple soft errors can be efficiently resolved using these codes.

9.3 COMBINATORIAL REPRESENTATION MATRICES

HSIAO Codes: Single error correction and Double error detection (SEC-DED)

For larger Hamming weights, linear SEC-DED codes used for memory protection fail to detect and sometimes wrongly correct many errors. Multiple bit errors become more common with large memories for new technologies and linear error correcting codes do not safeguard the system reliability. It is possible to improve upon the conventional Hamming SEC-DED codes to achieve enhanced error detection capability.

9.3.1 Optimal odd-weight column codes:

The minimum hamming weight required for single error correction and double error detection is 4. This means that a few columns of the H matrix are linearly independent. In order to meet this condition, the columns of the H-matrix should satisfy the following requirements: [29]

i) “There are no all-O columns.

ii) Every column is distinct.

iii) Every column contains an odd number of 1's (hence odd weight).”

First two constraints are satisfied by a distance-3 code. Fourth constraint requires the resulting code to have a minimum distance 4. The logic behind is that the modulo two addition of any odd number of odd-weight vectors always results in an odd-weight vector, and modulo-two addition of any even number of odd-weight vectors always results in an even-weight vector which includes the weight-0 vector. This is particularly useful for double error detection. Furthermore, the total number of 1's in every row of the Hamming matrix is proportional to the number of logic levels needed for the check bit and syndrome bit generation of that row. For instance, if the total number of 1's in the ith row is t_i , and the check bit and syndrome bit corresponding to the ith row of the H-matrix are C_i and S_i respectively, then,

$lc_i = [\log_v(t_i-1)]$

$ls_i = [\log_v(t_i)]$

Where,

lc_i id the number of logic levels needed to generate the checkbit C_i if a v-input modulo-2 adder is used,

ls_i is the number of logic levels needed to generate the syndrome bit S_i if a v-input modulo-2 adder is used,

and [X] is the smallest integer greater than or equal to X.

Number of inputs for the modulo-two adder is generally fixed. Hence, to reduce ls_i, t_i has to be reduced to its minimum value. If all values of ti (i = 1,2,...,r) are minimum and equal, the process of encoding and error detection in decoding, the two most critical processes in the memory operation, will consume very less time. This is advantageous also because it requires less hardware for implementation, which naturally implies higher reliability. Hence, minimizing t_i for all i is important from implementation perspective. [29]

9.3.2 Construction Procedure:

The construction procedure can be very clearly understood in terms of the parity-check matrix H. The columns to be selected from the H-matrix for a particular (n,k) code is based on the following criteria:

"1) Every column should have an odd number of l's; i.e., all column vectors are of odd weight.
2) The total number of 1's in the H-matrix should be a minimum.
3) The number of 1's in each row of the H-matrix should be made equal, or as close as possible, to the average number, i.e., the total number of 1's in H divided by the number of rows." [HSIAO]

These constraints originate from the logic described previously. If there are k data bits and r parity-check bits, the following equation has to be satisfied:

$$\sum_{\substack{i=1 \\ i=\text{odd}}}^{\leq r} \binom{r}{i} \geq r + k.$$

The number of check bits used by this code is same as that used by Hamming SEC-DED code. The H-matrix is constructed as follows:

1) "All $\binom{r}{1}$ weight-1 columns are used for the r check-bit positions.
2) Next, if $\binom{r}{3}$ >= 2 k, select k weight-3 columns out of all possible $\binom{r}{3}$ combinations. If $\binom{r}{3}$ < k, all $\binom{r}{3}$weight-3 columns should be selected. The leftover columns are then first selected from among all $\binom{r}{5}$ weight-5 column, etc. The process is continued until all k columns have been specified.
3) If codeword length n = k + r is exactly equal to $\sum_{\substack{i=1 \\ i=\text{odd}}}^{i\leq r} \binom{r}{i}$ " [29]

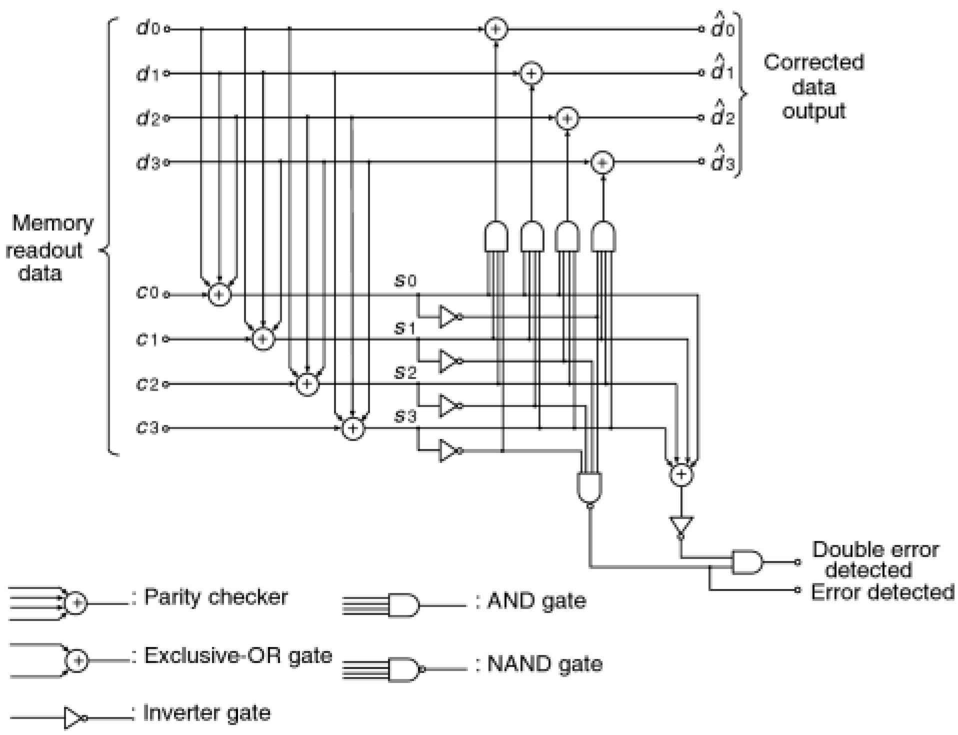


Figure47: Parallel Decoding Circuit [30]

Double error detection is achieved by considering the parity of all syndrome bits. If the syndrome bits have even parity, it implies that even number of errors has occurred. If there is an occurrence of multiple even errors, it is considered as double errors, because all errors are assumed to be statistically independent. This differs from the double error detection of Hamming Codes. In the case of hamming codes, occurrence of a single or double error is detected by making use of a special bit generated by an all-1 row in the H-matrix. Encoding and decoding process for error detection can be accelerated by eliminating the all-1 row in the H-matrix. In case of the parity check matrix, number of 1's in the H-matrix is much less than that in Hamming codes.

9.4 PRODUCT CODE:

Product codes are one of the well-known mechanisms employed for error correction since they provide protection against both random and burst errors. "Figure below illustrates a typical (n_v x n_h) array Γ over a field F = GF(q) that is encoded by a product code consisting of two linear codes: an [n_h, $k_h = n_v - r_v$, d_v] column code C_v over F." [Quote ref] This procedure of construction of product codes will be referred to as *Construction 0* going forward. The total redundancy encountered during the process is given by

$n_h r_v + n_v r_h - r_h r_v$

"A maximum distance separable code is a matrix which depicts a function, with certain diffusion properties. Technically, an m×n matrix A over a finite field K is an MDS matrix if it is the transformation matrix of a linear transformation f(x)=Ax from Kn to Km such that no two different (m+n)-tuples of the form (x,f(x)) coincide in n or more components." [wiki] In most of the applications, the codes C_h and C_v are considered to be maximum distance separable codes, for which, the equations $d_h = r_h + 1$ and $d_v = r_v + 1$ hold good. The requirement is that, the code lengths n_h and n_v should not exceed q+1. This condition would be met by default if the codes were to be naturally symbol-oriented, where the primary objective would be burst correction. Burst error or error burst is nothing but a contiguous sequence of symbols. Let us assume that the array transmitted is Γ. The traits of the burst errors are such that, the erroneous entries in the received array Γ' is limited to 'T' number of rows, where T is dependent on some probability measure, which is controlled by the channel and the choice of n_h and n_v. A corrupt row in the received array Γ' is the one that contains at least 1 erroneous entry. Assuming that the ith row in Γ' is affected, the corresponding error vector is obtained by the ith row of the error array E. Only if a row in Γ' is erroneous, the respective error vector is non-zero.

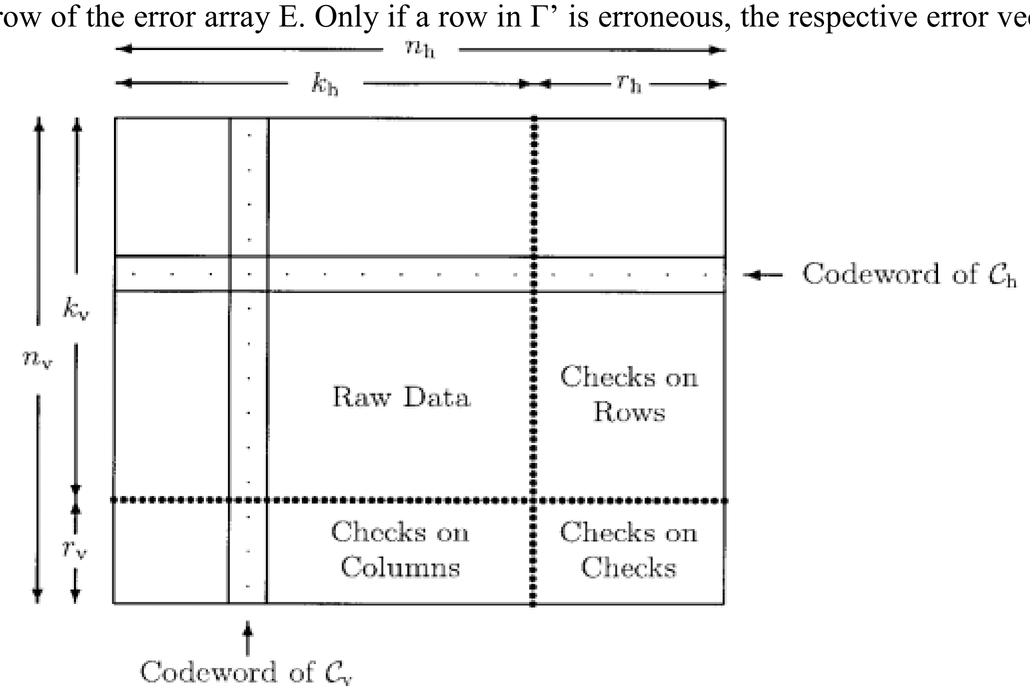


Figure48: Array of a product code [31]

Decoding technique:

First, the corrupted rows are loacted using the code C_h. Once the erroneous rows are found, coulmn by column application of the decoder of C_v is performed, considering the corrupt entries in each of the columns as ‘erasures’. Array decoding failure occurs when the originally encoded array is not accurately reconstituted by the decoder. Let ‘p’ be the accepted probability of array decoding failure. Half of this probability has to be kept aside for the case where the correction capability of C_v is exceeded by the number of errors. To be more specific, if C_v is an MDS code, then r_v can be considered such that

$\text{Prob}\{T> r_v\} <= p/2$. [31]

This is to ensure that the erasure correction capability of C_v is acceptable. Corrupt rows are detected by the code C_h by computing syndrome for each row, based on the corresponding C_h. Supposing the number of affected entries in a corrupt row is ‘l’, then l should be less than $d_h=r_h+1$, such that the syndrome calculated for that row must be a non-zero if the row is erroneous. In any parity check matrix of order r_h x n_h, every r_h column is linearly independent owing to C_h being MDS. The probability that all the syndromes of such a row being zero is q^{-rh}. Additionally, the probability of the row being corruptand having an all-zero syndrome is $q^{-rh}- q^{-l}< q^{-rh}$. Hence, the probability of misdetecting an erroneous row is less than q^{-rh}, number of affected entries in the erroneous row being inadvertant. This implies that the probability that a row is both misdetected and corrupt has to be

$\Sigma_t \text{Prob}\{T=t\}.t.q^{-rh} = \tau\, q^{-rh}$

Where τ is the expected value $E_T\{T\}$. This ensures that the overall probability of misdetecting a row will not be greater than p/2.

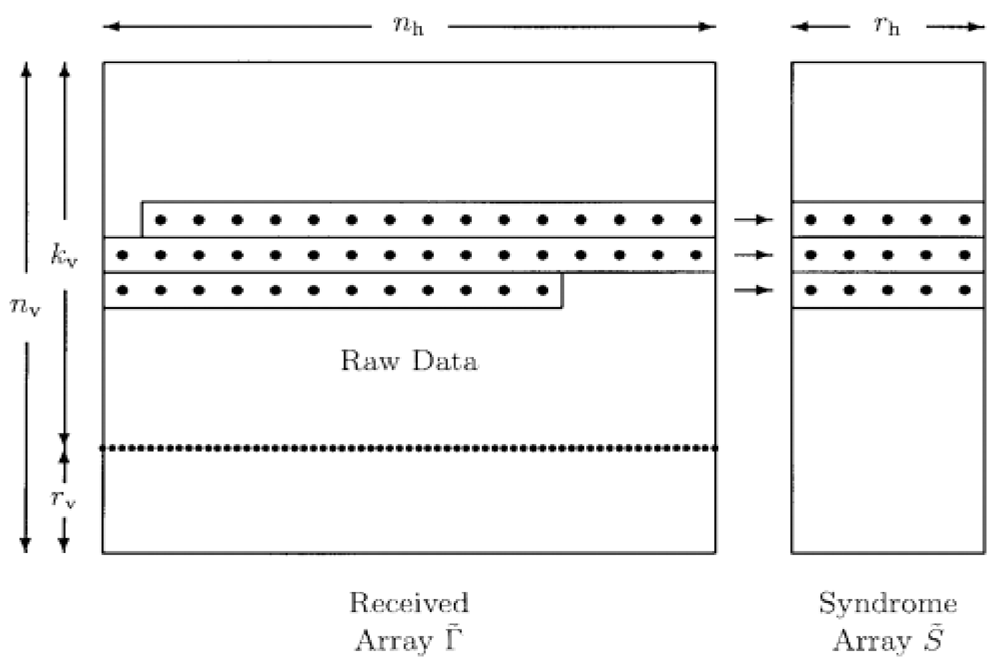


Figure49: Syndrome array for received array [31]

9.5 MATRIX CODE:

The name originates from the fact that the check bits are used in a matrix format. n-bits of data is divided into k_1 number of words of width k_2. A matrix of dimension $[k_1,k_2]$ is formed, where k_1 and k_2 represent the number of rows and number of columns respectively. One check-bit is added for each of these k_1 rows for single error correction and double error detection. An additional k_2 number of bits are added as vertical parity bits. The technique is explained in detail by taking example of a 16 bit word length memory.

The 16-bit data word is partitioned into a 4x4 matrix as illustrated in the figure below, so $k_1=k_2=4$. Each row is assigned with its corresponding Hamming code. For a data bit of length 4, number of check bits required for Hamming code are 3, hence, at the end of the data bits, 3 check bits are added.

X_1	X_2	X_3	X_4	C_1	C_2	C_3
X_5	X_6	X_7	X_8	C_4	C_5	C_6
X_9	X_{10}	X_{11}	X_{12}	C_7	C_8	C_9
X_{13}	X_{14}	X_{15}	X_{16}	C_{10}	C_{11}	C_{12}
P_1	P_2	P_3	P_4			

Table9: A 16 bit data word with check bits and parity bits

C_1, C_2, and C_3 are the Hamming check bits for the first row. Following equations are used to calculate the check bits:

$C1 = X2 \oplus X3 \oplus X4$

$C2 = X1 \oplus X3 \oplus X4$

$C3 = X1 \oplus X2 \oplus X4$

Where, Xi is ith bit of the data word. P1-P4, are the vertical parity bits and they are inserted as shown in Figure 1. Following equations are used for the parity bits generation.

$P1 = X1 \oplus X5 \oplus X9 \oplus X13$

$P2 = X2 \oplus X6 \oplus X10 \oplus X14$

$P3 = X3 \oplus X7 \oplus X11 \oplus X15$

$P4 = X4 \oplus X8 \oplus X12 \oplus X16$

Summarizing,

X_1 to X_{16} comprise the data bits, C_1 through C_{12} are the check bits, P_1-P_4 are parity bits.

Each row is decoded is using a hamming decoder. Two steps are used for decoding:

1) Syndrome bit calculation is done and data bits are used for check bits generation and the generated check bits are compared with the syndrome bits using modulo 2 addition $C_1 \oplus C_1' = S_1$. The method is called syndrome bit generation and S_1 is called the syndrome bit of the check bit '1'.

2) These syndrome bits are used to generate Single Error Detection (SED)/Double Error Detection (DED)/No Error (NE) signals for every row.

If the parity syndrome bits S_{pi} of every bit, and the saved value of the bit is used to generate DED signal, any single or double erroneous bits can be corrected using the equation below:

$X_{i\_correct} = X_i \oplus O_i \oplus DED_j \cdot S_p$ [31]

If more than two errors are present in the code word, two errors in each row can be corrected by matrix code assuming that each column has only one error. However, when there are only 2 errors, they can be corrected without any constraints.

10. CONCLUSIONS AND FUTURE WORK

Scaling in standard CMOS technology has been extremely advantageous in terms of speed of operation, lower power dissipation, and integration density. It has now been challenged by the problem of variability. An in-depth study and evaluation of low power techniques and how the system reliability is affected, was done through this thesis in four different levels. First, we studied the architecture and functioning of SRAM cell in detail, and various cell designs were proposed to enhance the performance and reliability. In circuit level, these proposed designs were simulated and evaluated by subjecting the cells under process variation. In the device level, the cells were realized by making use of different device models available to evaluate and compare their performances. Stability performances of the five SRAM cell designs have been presented. Finally, in the algorithm level, various error correction techniques were presented and assessed.

From all these analyses, we can conclude that, as the technology continues to advance, we will have faster SRAMs, but the devices will be more vulnerable to variations and mismatches, which degrade the static noise margin of SRAM cells. By isolating the read current path and by using transistors of minimum feature size, a greater immunity to process variations, along with a high density of on-chip memory can be achieved.

Although 6T SRAM is advantageous in terms of low static power dissipation and lesser area consumption on chip, it is less reliable when subjected to supply voltage reduction and process variation. 7T SRAM cell has the drawback of having 13% extra cell area as compared to the standard 6T SRAM cell. 7T cell design has a significantly larger SNM for a logic ‘0’ read operation (read SNM free feature), which improves the cell stability to a large extent, but the cell has a small write failure rate and poor read stability. The cell structure is quite simple and it can function well at low voltages. The read and write failure rates of the 8T SRAM cell are comparatively small, but the write stability suffers with the variation in temperature. If the read-disturb is not considered, the characteristic curves of the 8T SRAM cell is actually the voltage transfer curve of the two cross-coupled inverters, the SNM of which is definitely larger. Compared to the other designs, 9T SRAM cell has more stable read and write operations. It exhibits significant stability under process variation, though there is a trade-off of larger cell area, higher power consumption and a more complex cell structure. The cell structure of 10T SRAM cell is more complex, as compared to the 9T SRAM cell, but the read operation is less stable.

Multiple bit error correction using Hamming code has been implemented in order to achieve an ECC protected array of SRAMs. Other error correction techniques like HSIAO codes, Product Codes, and Matrix codes are investigated and evaluated.

Future Work:

Although this project focuses largely on dealing with degradation due to process variation, a bigger challenge lies in preventing the degradation process leading to permanent failures like radiation induced gate rupture where the damage is irreversible. This is an area where more research is needed to find a practical solution while determining the cost incurred for ensuring a high level of yield and protection. Also, the error correction technique described, to handle soft-errors in the memory, can still be extended for other error correcting codes like Reed-Solomon codes etc.

References

[1] Jawar Singh and Balwinder Raj “SRAM Cells for Embedded Systems”
[2] Ying Ying Zhang’s thesis “Low Power Process-Tolerant Design of Embedded SRAM Memory in Nano CMOS” 2011
[3]A. Islam, Mohd. Hasan ”A technique to mitigate impact of process, voltage and temperature variations on design metrics of SRAM Cell”
[4] Singh, J (Singh, Jawar)1; Mathew, J (Mathew, Jimson)1; Pradhan, DK (Pradhan, Dhiraj K.)1; Mohanty, SP (Mohanty, Saraju P.) "Failure Analysis for Ultra Low Power Nano-CMOS SRAM Under Process Variations"
[5] S Borker, T Karnik, S Narendra, J Tschanz, A.Keshavarzi, and V.De, “Parameter variation and impact on circuits and microarchitecture,” In 40th Design Automation Conference, June 2003, pp. 103–113.
[6] M. Yamaoka, K. Osada, and K. Ishibashi, “0.4-V logic-library-friendly SRAM array using rectangular-diffusion cell and delta-boosted-array voltage scheme,” IEEE J. Solid-State Circuits, vol. 39, no. 6, pp. 934–940, Jun. 2004.
[7] D.E. Hocevar, P.F. Cox, and P.Yang, “Parametric yield optimization for mos circuit blocks,” IEEE Trans. Comput. Aided Des. Integr. Circuits System, vol. 7, pp. 645–658, 1988.
[8]T. Karnik, P. Hazucha, J. Patel, “Characterization of soft error caused by single event upsets in CMOS processes”, IEEE Trans. Dependable and Secure Computing, 9(2), 128-143, April 2004
[9]Kulkarni, JP (Kulkarni, Jaydeep P.)1; Roy, K (Roy, Kaushik)2 "Ultralow-Voltage Process-Variation-Tolerant Schmitt-Trigger-Based SRAM Design"
[10] Asenov, A. “Random Dopant Induced Threshold Voltage Lowering and Fluctuations in Sub-0.1 um MOSFET's: A 3-D 'Atomistic' Simulation Study” *IEEE Trans. On Electron Devices*, vol. 45, no. 12, Dec 1998.
[11] Rabaey, J.M. Process Variations In: EE241: Advanced Digital Integrated Circuits lecture notes in Electrical & Electronics and Computer Science, University of California at Berkeley, 2006.
[12]Liu, D. and Svensson, C. “Trading speed for low power by choice of supply and threshold voltages,” *IEEE Journal of Solid-State Circuits*, vol. 28, no. 1, pp. 10-17, January 1993.
[13] Aly, R.E.; Bayoumi, M.A. "Low-Power Cache Design Using 7T SRAM Cell," Circuits and Systems II: Express Briefs, *IEEE Transactions* on , vol. 54, no. 4, pp. 318-322, April 2007.
[14] Sil, A.; Ghosh, S.; Bayoumi, M. "A novel 8T SRAM cell with improved read-SNM," Circuits and Systems, 2007. NEWCAS 2007. *IEEE Northeast Workshop on* , pp. 1289-1292, 5-8 Aug. 2007.
[15] Zhiyu Liu; Kursun, V.; "Characterization of a Novel Nine-Transistor SRAM Cell," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol. 16, no. 4, pp.4 88-492, April 2008.
[16] Ik Joon Chang; Jae-Joon Kim; Park, S.P.; Roy, K. "A 32kb 10T Subthreshold SRAM Array with Bit-Interleaving and Differential Read Scheme in 90nm CMOS,"*Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International* , pp. 388-622, 3-7 Feb. 2008.
[17] Aditya Bansal *et al*., “Impacts of NBTI and PBTI on SRAM Static/Dynamic Noise Margins and Cell Failure Probability”
[18] Shen-Fu Hsiao; Chia-Sheng Wen; Ming-Yu Tsai; Ming-Chih Chen. "Automatic generation of high-performance multiple-input XOR/XNOR circuits and its application in Advanced Encryption Standard (AES)," *Next-Generation Electronics (ISNE), 2010 International Symposium on* pp.77-80, 18-19 Nov. 2010
[19] Hsin-I Liu, “On-Chip ECC for Low-Power SRAM Design”

[20] Horiguchi, M. “Redundancy techniques for high-density DRAMs,” Innovative Systems in Silicon, 1997. Proceedings. Second Annual IEEE International Conference on , pp. 22-29, 8-10 Oct 1997. 19
[21] H. L. Kalter, *et al*. “A 50-ns 16-Mb DRAM with a 10 ns data rate and on-chip ECC,” IEEE Journal of Solid-State Circuits, vol. 25, pp. 1118–1128, Oct.1990.
[22] Agarwal, A.; Paul, B.C.; Mahmoodi, H.; Datta, A.; Roy, K. “A process-tolerant cache architecture for improved yield in nanoscale technologies,” Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol. 13, no. 1, pp. 27- 38, Jan. 2005.
[23] Andrei Pavlov “CMOS SRAM Circuit Design and Parametric Test in Nano Scaled Technologies
[24] IBM Applications Note “Understanding Static RAM Operation”
[25] http://en.wikipedia.org/wiki/Static_random-access_memory
[26] http://www.eeweb.com/project/joel_abraham/vlsi
[27] http://bwrc.eecs.berkeley.edu/classes/icdesign/ee141_s09/Project/EE141_s09_project1.pdf
[28] http://www.butingtech.com/explained-how-error-detection-and-correction-works/
[29] M. Y. Hsiao, “A Class of Optimal Minimum Odd-weight-column SEC-DED Codes,” IBM J. RES. Develop., Vol.14, pp. 395-401, July 1970.
[30] E. Fujiwara, Code Design for Dependable Systems, Theory and Practical Applications, Wiley-interscience, 2006.
[31] R. M. Roth and G. Seroussi, “Reduced-redundancy product codes for burst error correction,” IEEE Trans. Inform. Theory, vol. 44, pp. 1395–1406, July 1998.
[32] Error Correcting Codes, lecture slide, Professor Dhiraj.K.Pradhan
[33] http://en.wikipedia.org/wiki/Semiconductor_device_modeling
[34] H. Iwai “Roadmap for 22 nm and beyond”

APPENDICES

Appendix A

IMPLEMENTATION OF 6T SRAM CELL IN HSPICE:

```
*Simulation of 6T SRAM Cell
*definition
.PARAM
$ Number of sweeps for Montecarlo
+val=2000
$Load Capacitance
+cload=5pf
+vdd=1
$ Transistor sizing
+l=45n
+w=90n
$ Threshold voltage variation
*+nvt= 0.3423
*+pvt= -0.23122v
+nvt= agauss(0.342v,0.034v,3)
+pvt= agauss(-0.342v,0.034v,3)
$ Gate Oxide Critical Dimensions
*+ntox=1.25n
*+ptox=1.3n
*+ntox= agauss(1.25n,0.25n,3)
*+ptox= agauss(1.3n,0.26n,3)

.GLOBAL 1

*Initial Conditions
.IC V(Q)=0
.IC V(QB)='vdd'

VCC 1 0 'vdd'
Vcharge Charge 0 'vdd'
*---------------------------------------LOAD CAPACITANCES-----------------------------------*
CBL BL 0 cload
CBLB BLB 0 cload
*---------------------------------------PRECHARGE CIRCUIT------------------------------------*
XIPC PC PCB INV
MG2 BLB PCB Charge Charge PMOS W='18*w' L='l'
MG3 BL PCB Charge Charge PMOS W='18*w' L='l'
*-----------------------------------------VOLTAGES------------------------------------------------*
VW W 0 pwl 0n 0v, 29n 0v, 30n 'vdd', 50n 'vdd', 51n 0v,
```

```
+154n 0v ,155n 'vdd', 175n 'vdd', 176n 0v,245n 0v,R 0

VPC PC 0 pwl 0n 0v, 1n 'vdd', 21n 'vdd', 22n 0v, 69n 0v, 70n 'vdd',
+90n 'vdd', 91n 0v, 129n 0v, 130n 'vdd', 150n 'vdd', 151n 0v, 194n 0v,
+195n 'vdd', 215n 'vdd', 216n 0v, 250n 0v,R 0

VD D 0 pwl 0n 0v, 29n 0v, 30n 'vdd', 50n 'vdd', 51n 0v,
+154n 0v ,176n 0v,245n 0v,R 0

VWL WL 0 pwl 0n 0v, 39n 0v, 40n 'vdd', 50n 'vdd', 51n 0v ,100n 0v , 101n 'vdd',
+120n 'vdd', 121n 0v,164n 0v, 165n 'vdd',175n 'vdd',176n 0v,
+224n 0v , 225n 'vdd', 245n 'vdd', R 0

VDB DB 0 pwl 0n 0v, 154n 0v ,155n 'vdd', 175n 'vdd',176n 0v,245n 0v,R 0
*---------------------------------------6T NETLIST---------------------------------*
M1 Q QB 1 1 PMOS W='w' L='l'
M2 Q QB 0 0 NMOS W='1.2*w' L='l'
M3 QB Q 1 1 PMOS W='w' L='l'
M4 QB Q 0 0 NMOS W='1.2*w' L='l'
M5 Q WL BL 0 NMOS W='w' L='l'
M6 QB WL BLB 0 NMOS W='w' L='l'
*-----------------------------------------WRITE--------------------------------------*
MW BL W Qw 0 NMOS W='10*w' L='l'
MWB BLB W Qwb 0 NMOS W='10*w' L='l'
MRB Qwb D 0 0 NMOS W='10*w' L='l'
MR Qw DB 0 0 NMOS W='10*w' L='l'
*-----------------------------------------------------------------------------------------*
.SUBCKT INV in out1
*inverter subcircuit
MI1 out1 in 1 1 PMOS W='2*w' L='l'
MI3 out1 in 0 0 NMOS W='w' L='l'
.ENDS INV
*--------------------------TRANSIENT ANALYSIS--------------------------------*
.TRAN 0.1n 150n Sweep Vdd 0v 1v 0.2v

.meas tran pow AVG power
.option nopage nomod post
*--------------------------Computing read time--------------------*
.MEASURE ReadTime trig v(WL) VAL='0.5*vdd' Rise=2
+ targ v(BLB) VAL='0.9*vdd' fall=2
*--------------------------Computing write time-------------------*
.MEASURE WriteTime trig v(WL) VAL='0.5*vdd' Rise=1
+ targ v(QB) VAL='0.5*vdd' Fall=1
.option nopage nomod post
```

Appendix B

IMPLEMENTATION OF (7, 4) HAMMING CODE ALGORITHM IN HSPICE

```
$ECC (7,4) for 4 data bits
$Definition
.PARAM
+vdd=1.0v
+cload=5pf
+l=45n
+w=90n
+nvt= 0.46893
+pvt= -0.49158v
+nvt= agauss(0.469v,0.046v,3)
*+pvt= agauss(-0.469v,0.046v3)
+ntox=1.25n
+ptox=1.3n
*+ntox= agauss(1.25n,0.25n,3)
*+ptox= agauss(1.3n,0.26n,3)

.GLOBAL 1

$Initial Conditions
.IC
+V(Q1)=0
+V(QB1)='vdd'
+V(Q2)=0
+V(QB2)='vdd'
+V(Q3)=0
+V(QB3)='vdd'
+V(Q4)=0
+V(QB4)='vdd'
+V(Q5)=0
+V(QB5)='vdd'
+V(Q6)=0
+V(QB6)='vdd'
+V(Q7)=0
+V(QB7)='vdd'

VCC 1 0 'vdd'
Vcharge Charge 0 'vdd'
*------------------------------------------------------------------------------------------------------------*
*supply voltage source define

VPC1 PC1 0 pwl 0n 0v, 1n 'vdd', 21n 'vdd', 22n 0v, 69n 0v, 70n 'vdd',
```

```
+90n 'vdd', 91n 0v, 150n 0v,R 0

VPC2 PC2 0 pwl 0n 0v, 1n 'vdd', 21n 'vdd', 22n 0v, 69n 0v, 70n 'vdd',
+90n 'vdd', 91n 0v, 150n 0v,R 0

VPC3 PC3 0 pwl 0n 0v, 1n 'vdd', 21n 'vdd', 22n 0v, 69n 0v, 70n 'vdd',
+90n 'vdd', 91n 0v, 150n 0v,R 0

VPC4 PC4 0 pwl 0n 0v, 1n 'vdd', 21n 'vdd', 22n 0v, 69n 0v, 70n 'vdd',
+90n 'vdd', 91n 0v, 150n 0v,R 0

VPC5 PC5 0 pwl 0n 0v, 1n 'vdd', 21n 'vdd', 22n 0v, 69n 0v, 70n 'vdd',
+90n 'vdd', 91n 0v, 150n 0v,R 0

VPC6 PC6 0 pwl 0n 0v, 1n 'vdd', 21n 'vdd', 22n 0v, 69n 0v, 70n 'vdd',
+90n 'vdd', 91n 0v, 150n 0v,R 0

VPC7 PC7 0 pwl 0n 0v, 1n 'vdd', 21n 'vdd', 22n 0v, 69n 0v, 70n 'vdd',
+90n 'vdd', 91n 0v, 150n 0v,R 0


VWL WL 0 pwl 0n 0v, 29n 0v, 30n 'vdd', 50n 'vdd', 51n 0v ,100n 0v , 101n 'vdd',
+120n 'vdd', 121n 0v,150n 0v, R 0

VWL_P WL_P 0 pwl 0n 0v, 31n 0v, 32n 'vdd', 52n 'vdd', 53n 0v ,100n 0v , 101n 'vdd',
+120n 'vdd', 121n 0v,150n 0v, R 0

VWE WE 0 pwl 0n 0v, 31n 0v, 32n 'vdd', 52n 'vdd', 53n 0v, 150n 0v,R 0

VRE RE 0 pwl 0n 'vdd', 99n 'vdd' , 100n 0v,120n 0v, 121n 'vdd' ,151 'vdd',R 0

*-----------------------------Data Bits"0101100"--------------------------------------------------------*
**********1234567***********
VD3 D3 0 0
VDB3 DB3 0 pwl 0n 0v, 29n 0v, 30n 'vdd', 50n 'vdd', 51n 0v, 150n 0v,R 0 **************0

VD5 D5 0 pwl 0n 0v, 29n 0v, 30n 'vdd', 50n 'vdd', 51n 0v, 150n 0v,R 0***************1
VDB5 DB5 0 0

VD6 D6 0 0
VDB6 DB6 0 pwl 0n 0v, 29n 0v, 30n 'vdd', 50n 'vdd', 51n 0v, 150n 0v,R 0***************0
VD7 D7 0 0
VDB7 DB7 0 pwl 0n 0v, 29n 0v, 30n 'vdd', 50n 'vdd', 51n 0v, 150n 0v,R 0***************0
```

```
VSA SAE 0 pwl 119n 0v , 120n 0v, 121n 'vdd', 141n 'vdd', 142n 0v, 145n 0v, R 0
*---------------------------------------------------------------------*
**Pre-Charge ckt1**
XIPC1 PC1 PCB1 INV
MG21 BLB1 PCB1 Charge Charge PMOS W='18*w' L='l'
MG31 BL1 PCB1 Charge Charge PMOS W='18*w' L='l'
*---------------------------------------------------------------------*
**Pre-Charge ckt2**
XIPC2 PC2 PCB2 INV
MG22 BLB2 PCB2 Charge Charge PMOS W='18*w' L='l'
MG32 BL2 PCB2 Charge Charge PMOS W='18*w' L='l'
*---------------------------------------------------------------------*
**Pre-Charge ckt3**
XIPC3 PC3 PCB3 INV
MG23 BLB3 PCB3 Charge Charge PMOS W='18*w' L='l'
MG33 BL3 PCB3 Charge Charge PMOS W='18*w' L='l'
*---------------------------------------------------------------------*
**Pre-Charge ckt4**
XIPC4 PC4 PCB4 INV
MG24 BLB4 PCB4 Charge Charge PMOS W='18*w' L='l'
MG34 BL4 PCB4 Charge Charge PMOS W='18*w' L='l'
*---------------------------------------------------------------------*
**Pre-Charge ckt5**
XIPC5 PC5 PCB5 INV
MG25 BLB5 PCB5 Charge Charge PMOS W='18*w' L='l'
MG35 BL5 PCB5 Charge Charge PMOS W='18*w' L='l'
*---------------------------------------------------------------------*
**Pre-Charge ckt6**
XIPC6 PC6 PCB6 INV
MG26 BLB6 PCB6 Charge Charge PMOS W='18*w' L='l'
MG36 BL6 PCB6 Charge Charge PMOS W='18*w' L='l'
*---------------------------------------------------------------------*
**Pre-Charge ckt7**
XIPC7 PC7 PCB7 INV
MG27 BLB7 PCB7 Charge Charge PMOS W='18*w' L='l'
MG37 BL7 PCB7 Charge Charge PMOS W='18*w' L='l'
*---------------------------------------------------------------------*
CBL1 BL1 0 cload
CBLB1 BLB1 0 cload

CBL2 BL2 0 cload
CBLB2 BLB2 0 cload

CBL3 BL3 0 cload
```

```
M54 Q4 WL BL4 0 NMOS W='w' L='l'
M64 QB4 WL BLB4 0 NMOS W='w' L='l'
*-----------------------------------------------------------------------*
*6T SRAM_5 netlist
M15 Q5 QB5 1 1 PMOS W='w' L='l'
M25 Q5 QB5 0 0 NMOS W='1.2*w' L='l'
M35 QB5 Q5 1 1 PMOS W='w' L='l'
M45 QB5 Q5 0 0 NMOS W='1.2*w' L='l'
M55 Q5 WL BL5 0 NMOS W='w' L='l'
M65 QB5 WL BLB5 0 NMOS W='w' L='l'
*-----------------------------------------------------------------------*
*6T SRAM_6 netlist
M16 Q6 QB6 1 1 PMOS W='w' L='l'
M26 Q6 QB6 0 0 NMOS W='1.2*w' L='l'
M36 QB6 Q6 1 1 PMOS W='w' L='l'
M46 QB6 Q6 0 0 NMOS W='1.2*w' L='l'
M56 Q6 WL BL6 0 NMOS W='w' L='l'
M66 QB6 WL BLB6 0 NMOS W='w' L='l'
*-----------------------------------------------------------------------*
*6T SRAM_7 netlist
M17 Q7 QB7 1 1 PMOS W='w' L='l'
M27 Q7 QB7 0 0 NMOS W='1.2*w' L='l'
M37 QB7 Q7 1 1 PMOS W='w' L='l'
M47 QB7 Q7 0 0 NMOS W='1.2*w' L='l'
M57 Q7 WL BL7 0 NMOS W='w' L='l'
M67 QB7 WL BLB7 0 NMOS W='w' L='l'
*-----------------------------------------------------------------------*
*Generating Check Bits
*Synorome Generating Circuit
XC1 D7 D5 PC1 XORGate
XC2 D3 PC1 PC XORGate

XB1 D7 D6 PB1 XORGate
XB2 D3 PB1 PB XORGate

XA1 D7 D6 PA1 XORGate
XA2 D5 PA1 PA XORGate
*---------------------------------------------------------------------------------------*
*Input Data and Check bits being written into the SRAM CELLS
X1 PA WE BL1 BLB1 WriteP
X2 PB WE BL2 BLB2 WriteP
X3 D3 DB3 BL3 BLB3 Write
X4 PC WE BL4 BLB4 WriteP
X5 D5 DB5 BL5 BLB5 Write
```

```
X6 D6 DB6 BL6 BLB6 Write
X7 D7 DB7 BL7 BLB7 Write
*------------------------------------------------------------------------------------------*
*Data being read through the Sense Amplfier
XSA1 BL1 BLB1 SAE SON1 SO1 SA
XSA2 BL2 BLB2 SAE SON2 SO2 SA
XSA3 BL3 BLB3 SAE SON3 SO3 SA
XSA4 BL4 BLB4 SAE SON4 SO4 SA
XSA5 BL5 BLB5 SAE SON5 SO5 SA
XSA6 BL6 BLB6 SAE SON6 SO6 SA
XSA7 BL7 BLB7 SAE SON7 SO7 SA
*------------------------------------------------------------------------------------------*
XCretError SON3 SONB3 INV
*Detecing Error Bit Position
XEA0 SON6 SON5 PNA0 XORGate
XEA1 PNA0 SON7 PNA XORGate

XEB0 SON6 SON3 PNB0 XORGate
XEB1 PNB0 SON7 PNB XORGate

XEC0 SON5 SON3 PNC0 XORGate
XEC1 PNC0 SON7 PNC XORGate
*------------------------------------------------------------------------------------------*
.SUBCKT Write D DB BL BLB
*Write Circuit
MRB BLB D 0 0 NMOS W='10*w' L='l'
MR BL DB 0 0 NMOS W='10*w' L='l'
.ENDS Write
*------------------------------------------------------------------------------------------*
.SUBCKT WriteP D WE BL BLB
*Write Check Bits Circuit
MW BL WE Qw 0 NMOS W='10*w' L='l'
MWB BLB WE Qwb 0 NMOS W='10*w' L='l'
MWR1 DB D 1 1 PMOS W='2*w' L='l'
MWR2 DB D 0 0 NMOS W='1*w' L='l'
MRB Qwb D 0 0 NMOS W='10*w' L='l'
MR Qw DB 0 0 NMOS W='10*w' L='l'
.ENDS WriteP
*------------------------------------------------------------------------------------------*
.SUBCKT XORGate A B XOR
*XOR Gate NetList
MXOR1 2 A 1 1 PMOS W='4*w' L='l'
MXOR2 3 B 2 0 PMOS W='4*w' L='l'
MXOR3 3 B A 1 NMOS W='1*w' L='l'
```

```
MXOR4 3 A B 0 NMOS W='1*w' L='l'

MXOR21 XOR 3 1 1 PMOS W='w' L='l'
MXOR22 XOR 3 0 0 NMOS W='0.5*w' L='l'
.ENDS XORGate
*-----------------------------------------------------------------------------------------------*
.SUBCKT SA BL BLB SAE SON SO
*Sense Amplfier
MQ7 SO SAE 1 1 PMOS W='2*w' L='l'
MQ2 SO SON 1 1 PMOS W='2*w' L='l'
MQ1 SO SON Y 0 NMOS W='w' L='l'
MQ8 SON SAE 1 1 PMOS W='2*w' L='l'
MQ4 SON SO 1 1 PMOS W='2*w' L='l'
MQ3 SON SO YN 0 NMOS W='w' L='l'
MQ5 Y BL COM 0 NMOS W='2*w' L='l'
MQ6 YN BLB COM 0 NMOS W='2*w' L='l'
MQ9 COM SAE 0 0 NMOS W='5*w' L='l'
.ENDS SA
*-----------------------------------------------------------------------------------------------*
.SUBCKT INV in out1
*subcircuit of a inverter
MI1 out1 in 1 1 PMOS W='4*w' L='l'
MI3 out1 in 0 0 NMOS W='2*w' L='l'
.ENDS INV
*-----------------------------------------------------------------------------------------------*
*.TRAN 0.1n 150n Sweep monte=1000

.TRAN 0.1n 150n Sweep Vdd 0v 1v 0.2v
.meas tran pow AVG power
*-----------------------------Syndrome Bits Generation Delay------------------------*
.MEASURE STDelay trig v(D5) VAL='0.5*vdd' Rise=1
+ targ v(PA) VAL='0.5*vdd' Rise=1
*---------------------------------Error Detecting Time Delay-------------------------*
.MEASURE EDDelay trig v(SAE) VAL='0.5*vdd' Rise=1
+ targ v(PNC) VAL='0.6*vdd' Fall=1
*---------------------------------WriteTime--------------------------------------------*
.MEASURE WriteTime trig v(WL_P) VAL='0.5*vdd' Rise=1
+ targ v(Q2) VAL='0.1*vdd' Rise=1
.option nopage nomod post
```