TESTING AND FAULT ANALYSIS OF EMBEDDED SRAM IN LOW POWER TECHNOLOGIES USING PARASITIC R AND C EXTRACTION METHOD

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY
in
Electronics and Communication Engineering

By

Venkatesham Maddela

Registration Number: 41900665

Supervised By

Dr.Sanjeet K Sinha (22690)

SEEE (Professor)

Lovely Professional University

Co-Supervised by

Dr. Muddapu Parvathi

ECE (Professor)

BVRIT HYDERABAD College of Engineering for Women

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB
2024

DECLARATION

I, hereby declared that the presented work in the thesis entitled "**Testing and Fault Analysis of Embedded SRAM in Low Power Technologies using Parasitic R and C Extraction Method**" in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision of Dr.Sanjeet K Sinha, working as Professor, in the School of Electrical and Electronics Engineering of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)

Name of the scholar: Venkatesham Maddela

Registration No.: 41900665

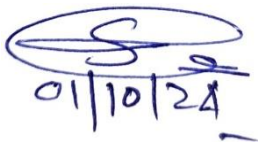
Department/school: ECE

Lovely Professional University,

Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled "**Testing and Fault Analysis of Embedded SRAM in Low Power Technologies using Parasitic R and C Extraction Method**" submitted in fulfillment of the requirement for the award of degree of **Doctor of Philosophy (Ph.D.)** in School of Electrical and Electronics Engineering is a research work carried out by Venkatesham Maddela, 41900665, is bonafide record of his original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.



(Signature of Supervisor) **(Signature of Co-Supervisor)**

Name of supervisor: Dr. Sanjeet K Sinha

Designation: Professor

Department/school: SEEE

University: Lovely Professional University

Name of Co-Supervisor: Dr.M.Parvathi

Designation: Professor

Department/school: ECE

University: BVRIT HYDERABAD
College of Engg. For Women

ABSTRACT

According to the International Technology Roadmap for Semiconductors (ITRS), memory occupied up to 94 percent of a SOC space. As a result, memory components dominate the fundamental logic of the SoC and take up greater space in Systems-on-Chip (SoCs). As process technology continues to scale, the test quality, yield, and dependability of current System-on-Chips increasingly rely on their embedded SRAM blocks. Their high integration density, along with a complex manufacturing process, results in minor lithographic flaws that necessitate increasingly effective test solutions to ensure low DPPM (defective parts per million) rates while keeping test costs low. So memories become the key distractor on SoC. As a result, in order to attain the best density and access speed possible, memories must be built independently of technological variance. The majority of existing fault approaches do not address manufacturing-related flaws. Because of its great density and fabrication using very deep submicron (VDSM) technology, testing embedded memory in a chip can be extremely difficult.

SRAMs, which can be utilized with SoCs or as a separate device, have become an essential component of memory in recent years. Because memories are employed in a variety of applications, the number of SRAM cores on SoCs is rapidly expanding. This results in increased density and die size. As with any other type of memory device, technology scaling reduces the size of integrated circuits and increases the density of transistors, resulting in undiscovered flaws. As a result, identifying and troubleshooting SRAM problems is challenging, and efficient test procedures are required. The difficulties in testing SRAMs include test time and fault coverage.

Prior to 1990, fault models like as AC parametric testing and DC parametric testing were utilized to test memory. Static faults are simple flaws that occur in SRAM cells. With a single read or write action, static defects can be exposed. March tests are novel fault models developed based on computational procedures. Electrical fault models are utilized

at the cell level, while Inductive Fault Analysis (IFA) is used at the layout level.
Because testing the memory for all faults is impractical, testing is limited to a subset of defects. This type of fault is referred to as a fault model. Physical spot flaws, which are represented as local irregularities in an SRAM's layout, are utilized to create a fault model for SRAMs. Many errors occur during the production process. Later, based on the sequence of read and write operations, additional test models called as March tests were established. The majority of existing fault models were assessed in terms of well-known March tests, which only provide fault detection information. March testing are effective for checking that SRAMs work properly. With these methodologies, as the fault coverage improves, so does the testing complexity and time. Other faults, termed as dynamic faults, take repeated read and write operations to sensitize. Resistive Dynamic faults are caused by open or bridges. These failures occur in the SRAM cell's address decoder, sense amplifiers, and pre charge circuits (bit lines). These defects are mostly responsible for the cell's electrical failure.

The linked faults occur in two or more cells. It was discovered that there are extremely few studies on coupling problems accessible. The fundamental reason for this is that when more than one cell is examined, the fault number is either doubled or tripled when compared to single cell faults. Because of this flaw magnification, utilizing march algorithms necessitates higher number primitive compositions, which takes a long time. Scaled-down technologies have an impact on parasitic effects, resulting in an extra source of defective behavior and making current test algorithms vulnerable to them. Traditional testing methods do not cover all flaws. To detect undefined defects, a fault model that takes the parasitic effect into account must be developed.

In order to accomplish this, the author developed a layout dependent Extraction of Parasitic R and C approach for fault detection and location identification. A new fault model for SRAM is provided, in which the defective model manifests as local disruptions in the SRAM cell layout. Three technologies are being considered: 45nm, 32nm, and 7

nm. The proposed test approach yielded 100% fault coverage. The test findings for variation levels ranging from Deep submicron (120nm) to Very deep submicron (7nm) are tabulated and examined. The parasitic variations are compared to the parasitic variations of fault-free SRAM. The proposed parasitic extraction method determines the type of fault as well as its position regardless of the specified technology (45nm, 32nm, or 7nm).

Machine learning has now become an essential method for assessing Embedded Memory. Machine learning improves overall performance and power economy. Machine Learning techniques can also automate the defect identification process. The author employs multiple linear regression to forecast parasite R and C values, with different technologies and lengths serving as independent variables and parasitic R and C serving as dependent variables. The results demonstrate that the parasitic R and C values may be predicted with an accuracy of 88.62%. The decision tree machine learning approach was utilized by the researcher to forecast fault detection and placement. regardless of technology variation, an excellent accuracy score of 91.78% in separating defective memory cells and locating the position of the problem.

Finally, a fault model dictionary that can be used for testing was developed for 45nm, 32nm and 7nm technologies for a single 6T-SRAM cell and multiple 6T-SRAM cells considering all open and short defects. Faults like Stuck at faults (SAF), Write Before Access Fault(WBAF), Undefine Write Fault(UWF), Undefined Read Fault(URF), Unstabilized Write Fault(USWF), Unstabilized Read Fault (USRF), No Access Fault(NAF), Transition Faults(TF), State Coupling faults(CFst), Inversion Coupling Fault(CFin), Fault Masking observed by this method and also we have observed an Undefined faults named as Undefined Short Fault(USF)

OBJECTIVES OF THE PROPOSED WORK

1. Analyze faults due to open defects in Embedded SRAM cell for submicron to deep submicron technologies
2. Design a parasitic R and C extraction method for short/bridge defects in SRAM cell using nano meter technologies.
3. Develop a fault detection method for open and Short defects in multi cell SRAM architecture
4. Design Novel Fault detection and test methods using Machine Learning Algorithms in embedded SRAM architecture

ACKNOWLEDGEMENT

During the pleasant journey of this thesis work, I have come across many people who have supported and motivated me all throughout my research work and thesis preparation.

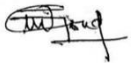
First of all, I would like to express my gratitude to my supervisor **Dr. Sanjeet Kumar Sinha** Associate Professor, School of Electronics & Electrical Engineering, Lovely Professional University (Phagwara, Punjab) for his constant support, encouragements, constructive discussions and critical analysis of the research topics. His positive attitude and belief have always motivated me to carry forward my research even during the days of my struggle. I am truly very fortunate to have the opportunity to work with him.

I would also like to thank **Dr. Parvathi Muddapu**, Professor, Electronics & Communication Engineering, BVRIT HYDERABAD College of Engineering for Women (Hyderabad, Telangana) for her guidance and support. I had many fruitful discussions about my work with her. Her scholastic discussions have encouraged me to work on my limitations and constantly improve my work.

I am highly thankful to **Mr. Vinay Sharma**, Director at ni2designs (Pune) for making all the necessary research related licensed softwares available at my disposal.

I am also thankful to all my colleagues of Electronics and Communication Engineering Department and my friends who contributed directly or indirectly through their constructive criticism in my research work.

I extend my deepest gratitude to my family members for their love, affection, encouragement and support during this whole journey.



Venkatesham Maddela

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ITRS	International Technology Roadmap for Semiconductor
SoC	System on Chip
VLSI	Very large Scale Integration
SRAM	Static Random Access Memory
eSRAM	embedded SRAM
DRAM	Dynamic Random Access Memory
MW	Microwind
DRC	Design Rule Check
DFT	Design For Testability
NAF	No Access Fault
BIST	Built In Self Test
MBIST	Memory BIST
VDSM	Very Deep Sub Micron
RAM	Random Access memory
ROM	Read Only Memory
SAF	Stuck at Faults
TF	Transition Faults
BF	Bridging Faults
ADF	Address Decoder Faults
NPSF	Neighborhood Pattern Sensitive Faults
DRF	Data Retention faults
FP	Fault Primitives
FFM	Functional Fault Model
SOF	Stuck Open Fault
RDF	Read Destructive Fault

DRDF	Deceptive Read Destructive Fault
IRF	Incorrect Read Faul
RRF	Random Read Fault
USF	Undefined Short Fault
CFst	State Coupling Fault
CFds	Disturb Coupling Fault
CFrd	Read Disturb Coupling Fault
CFdrd	Deceptive Read Disturb Coupling Fault
CFir	Incorrect Read Disturb Coupling Fault
CFrr	Random Read Coupling Fault
CFtr	Transition Coupling Fault
CFin	Inversion Coupling Fault
CUT	Circuit Under Test
BITE	Built in Test Equipment
ML	Machine learning
AI	Artificial Intelligence
CNN	Convolutional Neural Networks
DL	Deep Learning
EDA	Electronic Design Automation
CMOS	Complementary Metal Oxide Semiconductor
IFA	Inductive Fault Analysis
URF	Undefined Read Fault
UWF	Undefined Write Faults
SF	State Fault
SD	Short Defect
OD	Open Defect
USWF	Unstabilized Write Fault
USRF	Unstabilized Read Fault
WBAF	Write Before Access Fault
IOF	Initialization Order Fault

CHAPTER 1

INTRODUCTION

Integrated circuits (ICs) are main components of an electronic product. In integrated circuit fabrication channel length has been rapidly decreased. These reduced channel length devices circuits face many challenges during manufacturing and during functional working. The manufacturing with reduced channel length for reduce integrated circuit size, reduce power dissipation and low cost, create various defects in SRAM's core cell. These defects create various faults in SoC's main part SRAM. So, testing is necessary parts during manufacturing of integrated circuit for improve reliability for customer satisfaction and error free operation of SRAM in very sensitive and security system.

In the last few decades, size of circuits has decreased enormously and transistors on a System on-Chip (SoC) have increased from few thousands to billions in the time frame of three decades. Exponential increase in transistor count on a silicon chip is possible due to technology scaling. Density of a SoC increased beyond billions of gates which have resulted in extensive complex very dense integrated circuits. Due to this complexity of VLSI circuits, testing is also very complex, formidable, time and power consuming task. Eventually, test time and power consumption is increasing day by day and so, now the testing cost is very high and testing is an important factor in SoC or embedded system fabrication. Efficiency, test time and power consumption are main factors of testing. Efficiency increases the customer satisfaction, low test time and low power requirement reduces the cost of product. Therefore, in highly competitive and volatile market testing is key factor for survival of manufacturer, supplier and distributer.

The semiconductor part of integrated circuit used for fabrication of memory is regularly increases in latest nanometer technologies. A memory density will approach more than 90 per cent of System on Chip (SoC) semiconductor area in the duration of next decade [1] having its own field of applications. Static Random Access Memories (SRAMs) have wide use in embedded system and processor for fast processing with very low power consumption. SRAMs are very low power consumption volatile memories. SRAM stored data in core cell is accessed

with same and high speed. SRAM losses the data when power is off but with continuous power supply stored data remain saved and any time core cell can be accessed for functional operation. The term ‘static’ refers to the structure of six transistor storage core cell. Various resistive open defects and other variations during the manufacturing effect the operation or accuracy of SRAMs. So, memories have to be tested correctly or with zero error. SRAM testing challenges are represent fault model, high fault coverage, low power consumption and low time consumption test solution. . Different test methodologies have been evolved to identify the memory defects. Traditional test methods are zero/one, checker board, GALPAT, walking 1/0, sliding diagonal and butterfly to name a few. Among the traditional test methods, few methods are simple (zero/one, checkerboard) and few methods are complex (GALPAT, walking 1/0, sliding diagonal and butterfly). However, simple methods have poor fault coverage and the complex methods exhibit slow performance in spite of better fault coverage [2, 3 and 4]. Now in latest nanometer technologies these test solutions are not sufficient for new faults known as dynamic faults [5, 6 and 7]. These dynamic faults can be detected or sensitized by more than one read or write operations or Read Equivalent Stress (RES). Hence March algorithm with large operations element, enhanced addressing sequence and power constrained test schedule with low power test architecture will detect these faults with low power and minimal application time [8]. The complexity of the March algorithms increases as the fault coverage increases. This is the main disadvantage of the March Algorithms.

1.1 Semiconductor Memories Challenges

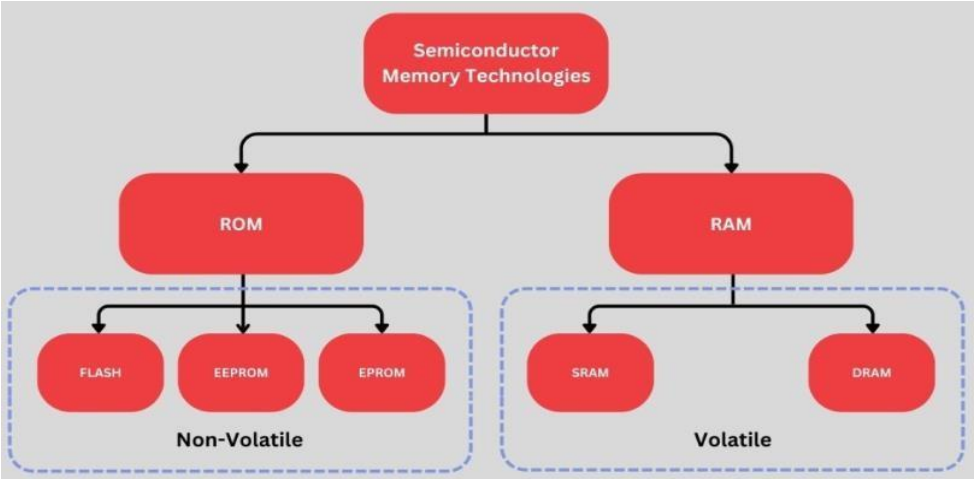


Figure. 1.1 Types of Semiconductor Memories

Semiconductor memories uses the semiconductor devices to store the data. There are two types of semiconductor memories volatile and non-volatile memories. Volatile memory refer to the device that loose the stored information when the supply voltage is turned off, while the non-volatile memories, or Read Only Memories(ROM) able to maintain the stored information even without an external power source. Volatile memories can be classified into two types based on the technology used to store the information Ex: SRAM and DRAM

Static Random Access Memory is a volatile memory, commonly used with embedded systems. In SRAM flip flops are used to store the data, SRAM can hold the data as long as power is ON. SRAM memory provides high-speed access and consumes little power, making it excellent for quick data access and temporary storage. The main disadvantages of SRAM are its complexity and high manufacturing expenses.

Another type of volatile memory that is often utilized in embedded systems is dynamic random access memory (DRAM). Unlike SRAM, DRAM use capacitors to store the data. Therefore DRAM must be refreshed regularly to keep their charge. DRAM has a lesser data lifetime and slower access time than SRAM due to the refresh required. The fundamental disadvantage of DRAM is its slow access time and periodic refreshing.

Depends on the application, memories classified into two types. i) Embedded Memories ii) External (Standalone) Memories.

1.1.1 Embedded Memories:

The memory integrated with core logic is known as embedded memory. This type of memory used to perform specific functions. Embedded memory enables the logic core to perform its functions without requiring inter-chip communication. Because of high speed and wide bus capabilities of embedded memory become an important component in VLSI. The capability to integrate memory and logic on the same chip, as well as advances in manufacturing technology, have made developing embedded memory devices easier.

Memory types for embedded systems are critical components in the design of efficient and effective systems. Memory selection can have an impact on performance, cost, and power usage. Each memory type has distinct features that make it appropriate for specific applications.

Because of its quick access time, SRAM is frequently employed in embedded systems for essential data pathways and cache memory. SRAM is also frequently used in conjunction with other memory types, to achieve a maximum performance and cost. Microcontrollers, digital signal processors, and high-speed data buffers are examples of embedded memories.

Factors to Consider to choose the memory in Embedded System

1. High access Time of the embedded system, time taken to read or write the data into the memory is known as access time
2. Cost of the memory should be low, because it will affect the total cost of the embedded system
3. Data Retention: the ability to hold the data, when the power is turned off known as data retention. Non-volatile memories keep the data even after power is turned off.
4. Density and Capacity

Memory density and capacity are critical for embedded systems that need to store significant volumes of data. Flash memory is an appealing alternative for such systems because it offers high-density storage at a low cost. Because of its greater cost-per-byte, SRAM delivers lesser density.

In conclusion, while choosing the memory for Embedded System, designer should have the knowledge on how to optimize the performance, reduction of power consumption and cost

1.1.2 External Memory Devices

Memory devices located external of the logic core are referred to as external memory devices. Embedded SRAM (Static Random Access Memory) and ROM (Read Only Memory) are currently commonly used. External memory devices, on the other hand, are freestanding memory devices such as hard disks and RAM that are not included on the chip.

External memory devices are memory devices that are not built inside a chip. These are devices like as hard drives, CD/DVD ROM, RAM, and ROM that are not built into the chip. Historically, external memory refers to devices used for permanent storing of huge amounts of data, such as magnetic disks, CD ROM, and so on. The most common external memory device is the hard disk, which can often store a huge quantity of data.

1.1.3 Embedded Memories Vs External (Stand Alone) Memories

Embedded memory devices are those that are integrated with the logic core on the semiconductor, and external memory devices are those that are located outside the chip. Outside or stand-alone SRAM and ROM are used less frequently than embedded SRAM and ROM. The use of embedded memory devices reduces the number of chips and the device's space requirements. Furthermore, on-chip memory delivers faster response time and reduced power consumption than external memory devices. Creating embedded memory devices, on the other hand, demands a more complex design and manufacturing process than creating external memory devices. Combining many types of memory on the same chip would also complicate the manufacturing process.

Memory size, density, speed, stored data accuracy and efficiency in ‘read’, ‘write’ operations in present VDSM (Very Deep Sub Micron) technology completely affects the working performance of various embedded systems and System on Chips related to the ‘audio’, ‘video’, data processing applications. Semiconductor Industry Association analysis forecasts that memory will cover more than 98 % semiconductor

area of a SoC or in a system's Integrated Circuit. Basis on these reasons technology limits are used to achieve maximum density, highest speed and accuracy in operations of the system.

The testing of SRAM is one of the manufacturing steps that ensure that the SRAM, manufactured has no manufacturing defect. The testing of SRAM's core cell working is very important step because it detects faults and cause of these faults due to various defects with electrical equivalent is resistance. Therefore efficient test operation improves the quality and reliability of integrated circuit products with maximum customer satisfaction [9, 10, 11 and 12]. Physical testing will help to eliminate errors, so working performance will enhance afterward. Fault free device efficiency is maximum and improved performance. Testing detects faults generated by defects and also gets results by analysis for reason of faults and defects in core cell during manufacturing. Testing improves the performance at all stages such as designer, production, test, manufacturer and end-user [13]. Memory core cell array bit line capacitance and other lumped capacitances are high and significant during process of read and write operations on which this study will be focusing.

1.2 Need of Memory Testing

Gordon Moore, predicted in 1965 that for every, one and half year it can be possible to integrate double the number of transistors. The term VLSI was used in 1980's for the ICs integrated with more than one lakh transistors. Now we are able, to integrate millions and billions of transistors in an integrated circuit. As a result, transistors are scaled down and interconnects are scaled down up to Nano meters. The scale down of transistors lead to high clock speeds [14]. There by high speed systems are designed. The result of scaling down of transistors lead to manufacturing defects in SoC [15]. Defects in the manufacturing process lead to faults in the chip. These VLSI chips used to construct an embedded system also behaves unexpectedly. A fault in the chip may lead to malfunctioning of the system. So, testing of memory cells is very much required. The expense of detecting a fault becomes ten times in the ratio when it is ignored from component stage to chip stage, chip stage to circuit board stage, circuit board stage to system stage and finally from system stage to the field of work stage. This rule is called as rule of ten.

A circuit deformity is treated as a fault, the effect of a fault is the malfunctioning of

the circuit, and a circuit malfunctioning can bring about a system failure. Two noteworthy deformity systems may be responsible to SOC configuration to breakdown. First are the manufacturing errors and the second are soft errors. During assembling, the manufacturing faults are observed. They are physical in nature that reasons the outline to neglect to work appropriately in the device, on the PCB, or in the complete logic block. Static faults are because of these manufacturing defects in the system. Stuck at faults and timing issues are come under this category. The rule of ten depicts that the cost of distinguishing a faulty device increments by a factor of size as we move through each phase of assembling, from scrap level, to board level, to system level, lastly to device assembled in the field.

1.3 SRAM Core Cell Operation

VLSI memory design began when the IBM and Intel Corporation came up with their invention, semiconductor memory in 1970. Till then magnetic thin films are used for storage purpose. Semiconductor memories replaced magnetic thin films. A drastic shift has taken place within no time. Semiconductor memories can be broadly differentiated as Random-Access Memory (RAM) and Read Only Memory (ROM).

Despite of its high cost, SRAM has many features like its high speed, very easy to use. So, it is used in almost all personal computers, super computers, main frame computers, embedded systems and handheld devices. Figure 1.1 illustrates the schematic diagram of a standard SRAM circuit. SRAM circuit constructed using a flip flop placed in between power supply lines, and two switching transistors.

Data is written into also read from the memory by using BL and BL_B lines as shown in figure. A write operation is carried out by supplying two inverting inputs al BL and BL_B. If it is a logic ‘1’ write operation. HIGH logic is supplied at BL and LOW logic is supplied at BL_B. If it is a logic ‘0’ write operation logic LOW is supplied at BL and logic HIGH is supplied at BL_B. A read operation is carried by retrieving information from BL and BL_B. There is no continuous refresh is required in the case of SRAM.

A standard SRAM memory cell is shown in Figure 1.1. Two NMOS and two PMOS transistors shapes the basic flipflop to store the information and word line (WL) is used to control, two NMOS pass transistors to excite bit line (BL) and bit line complement (BL_B) into the memory block. A compose activity is done by charging the bit line (BL) and bit line complement (BL_B) to which the logic level needs to be stored in to the SRAM cell. Making the word line (WL) at logic high plays out the memory write task, and the new information is written in the SRAM. In the same way, read task is started by bit line (BL) and bit line complement (BL_B) pre-charged to logic high level. The Word line (WL) put at logic high level to short NMOS pass transistors to set the value written in the cell on the bit line (BL) and bit line complement (BL_B).

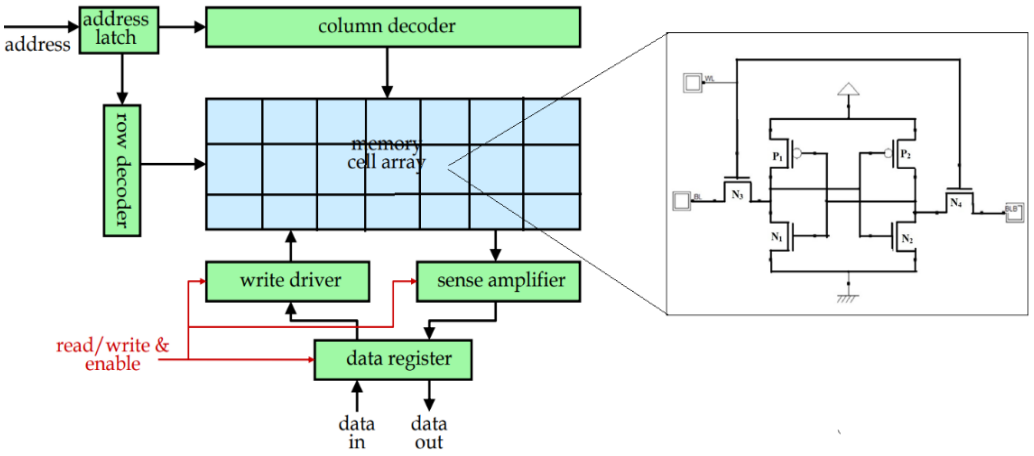


Figure 1.2: 6T SRAM memory cell

In Figure 1.2 a column single core cell of SRAM using six transistors with lines ‘BL’, ‘BLB’ and ‘WL’. It consists of a bi-stable multi-vibrator unit made from two cross coupled CMOS inverters and two ‘nMOS’ pass transistor. These ‘nMOS’ pass transistors connects ‘bit line’ (BL) and ‘bit line bar’ (BLB) according to control signal ‘word line’ (WL) instruction. The level if bit line and bit lines bar is changes according to operation performing. During ‘read’ operation voltage difference in bit line and bit line bar gives the information about saved data.

To furnish the details for the read operation of SRAM cell, let consider a memory cell storing a logic value '1"with logic node S at supply voltage V_{DD} as well as logic node SB at 0V. In read operation, the BL and BLB pins are pre-charged with certain supply voltage value V_{DD}. If the word line signal WL is connected to V_{DD}, then access transistors M_5 and M_6 are turned into ON state. Then there S node and bit line BL are in at the same potential i.e. V_{DD}, and there is no current flow takes place. There is a current flow from BLB (precharged to V_{DD}) through transistor M_6 and transistor M_4 by evoking the discharge of BLB. The charging and discharging path of the SRAM circuit during a read operation given in the Figure 1.3(a), where the bit line capacitor C_{BL} and bit line_bar capacitor C_{BLB} are equivalent capacitances of BL and BLB pins

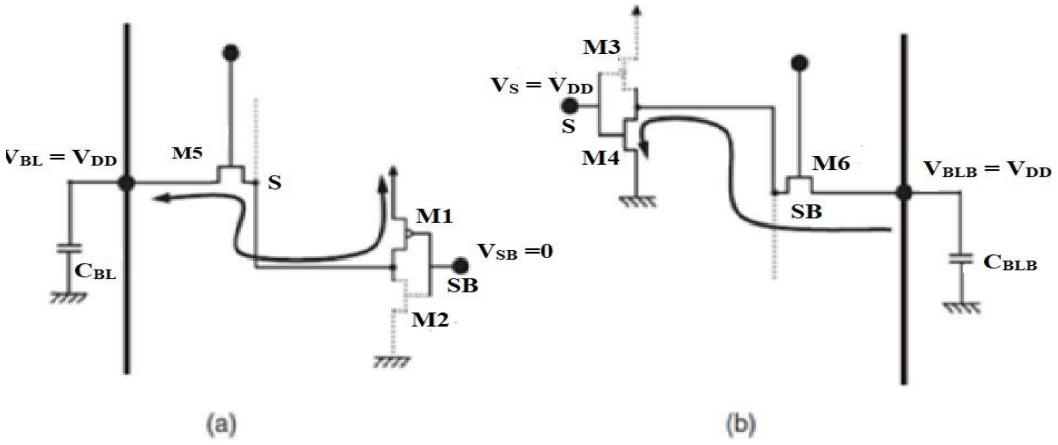


Figure 1.3(a): Read Operation of SRAM Cell

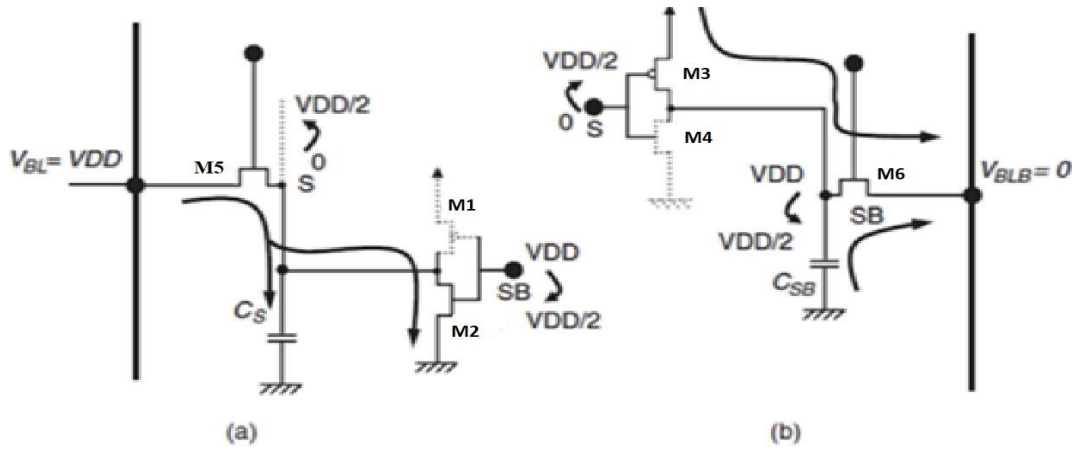


Figure 1.3(b): Write Operation of SRAM Cell

In the Figure 1.3(b), the capacitors connected at output nodes S and SB of two inverters CS as well as CSB were the parasitic capacitances. The regenerative feedback that causes the Flip-Flop to switch when either VS or VSB reaches to the value VDD/2. If this condition was satisfied, the positive feedback in this SRAM immediately takes place. This positive feedback establish a path to charge and discharge VDD in capacitors C_S and C_{SB}.

It is important to test the fabricated IC thoroughly. ITRS demonstrates that as size of memory increases, yield decreases which points out the significance of memory testing. Moreover, memory testing must have diagnostic capability to identify the defective locations. The circuit design and the manufacturing process can be improved significantly by unravelling the defective location. The results of fault diagnosis can further be used to examine the defect and the mechanism of failure. Hence memory testing plays a crucial role in semiconductor manufacturing process as it analyses and isolates many of the defects in order to obtain an efficient yield.

The main applications of the eSRAM is i) Automotive, mobile devices, consumer devices, media processors, and analog and mixed signal designs. ii) used to store firmware or security code iii) media applications iv) space craft applications

1.4 Faults and Fault Models

1.4.1 Assertion Faults: Wang et al. presented these assertion faults to depict the faults in embedded processors [20]. Test generation and validation of embedded array blocks are represented with the faults occurring in those circuits. Test generation from the abnormal state declaration determination gives a perfect answer for the issue of both test and verification. On the off chance that tests can be derived specifically from assertions, DFT on the gate level test point of view is never again required and consequently, dispose of the diagnosis needs connected with it. This additionally stays away from the inconveniences of experiencing the ATPG development tools. Hence, rather than making complex gate level perspectives for microprocessor arrays with the goal that they can be tested through ATPG tools, the new stream uses the high-level assertion.

1.4.2 Behavioural Faults: At the point where a computer programming language is

used to depict an electronic system, which is for the most part described by any programming language, just like C or java or any Hardware Description Language (HDL) that takes after a programming language, for example, VHDL or Verilog HDL. At the behavioural level, refered as practical or functional level state, the descriptions depicted are not the representing the electrical nodes and their interconnections but rather they simply define the functional behaviour of the system. Behavioural faults refer the inaccurate compilation of the statements and builds utilized as a part of the description. Cases of behavioural faults may come under declaration faults, instruction command faults and branch faults. In behavioural level of description, programming test strategies, statement scope and branch scope and toggle scope, are additionally utilized in spite of the fact that these don't fit in with a particular fault model.

1.4.3 Branch Faults: Branch faults are displayed in the behavioural level where a system is depicted with a programming language or hardware description language. Branch faults may occur because of a circuit branching to an unintended branch [21-22].

1.4.4 Bridging Faults: This type of faults mainly because of some of the circuit wires may be short circuited. In bridging fault with respect to memories, one cell alters another cell, there is no question of one cell dominating other. Generally demonstrated a bridging fault is because of the short between two or more wires in a circuit. The logic estimation of the shorted net might be displayed as OR bridging fault or AND bridging fault or Indeterminate. Combinational bridging faults and the scope of occurrence is generally more when appeared along with stuck at faults. That isn't generally appeared in feedback bridging faults which deliver stored values in the combinational logic [23-24]. These bridging faults are generally because of defects in the circuit construction.

1.4.5 Cross point Faults: Cross point faults generally displayed in Programmable Logic Devices (PLD). While designing a PLD, general input wires and output wires are crossing the product signals. Intersection flag lines either frame shorts of connections or stay open at intersection nodes, contingent upon the logic which is implemented. These cross-point discrepancies can be broad classified in two ways.

First is the missing cross point implying a disconnected association at an intersection where an association was expected. Second is the additional cross point implies a defective association at an intersection where no association was expected [25-26]. In view of their effect in the behavioural capacity of the PLD, the cross-point faults are additionally named shrinkage faults, development faults, appearance faults and disappearance faults.

1.4.6 Defect Oriented Faults: Unintended contrast between the fabricated system and the proposed structure of the system is called a defect. [3] in their research work mentioned about this type of defect-oriented flaws. Issues in the physical manufactured chip that for the most part happen during construction are called as defects. Physical deformities may create the electrical or logic level faults. Faults because of these physical deformities are called as defect oriented faults. Cases of physical deformities may be some open circuited wires, short circuits, bridging between adjacent wires, inappropriate semiconductor doping, and wrong components. Bridging faults, stuck-open faults are some examples of defect oriented faults.

1.4.7 Delay Faults: Delay faults in a system are because of combinational delay. If the combinational delay exceeded by the clock period, the system functionality may be changed. Different delay faults noted are gate delay faults, transition faults, segment delay faults, line delay faults and path delay faults[9].

1.4.8. Pattern Sensitive Faults: Since memory is firmly coupled neighboring cells influence the conduct of a cell. The example that is put away in the influencing cells modifies the conduct of influenced cell. This is called Pattern Sensitive Fault[13]. There are two kinds of pattern sensitive faults. One is Active PSF and the other Passive PSF. Dynamic PSF is the point at which an adjustment in any four contiguous cells supplements the incentive in the influenced cell. Passive fault is the point at which a cell is encompassed by four neighbouring cells and all the four cells at logic 0 level then the influenced cell stuck at logic zero level. Similarly, when a cell at logic 1 encompassed by four cells at logic 1 then the influenced cell stuck at logic 1. This kind of fault is called passive PSF.

1.4.9 Physical Faults: Physical faults in the circuits are because of the defects in

physical implementation of the circuits[31]. Cases of physical deficiencies are open circuited connections, short circuits between transmitters conveying detached flags, shorted or open transistors, and so forth. These deficiencies may be sometimes called as defect oriented faults.

1.4.10 PLA Faults: Faults observed in Programmable Logic Arrays (PLAs) are called as PLA faults. In PLA design, primary input and output lines are crossing the OR input lines. Convergence input lines and product lines may be outline required sorts of associations or may be open, dependent upon the executed digital logic. These Cross point discrepancies are classified in two ways. First one is the open circuited cross point where a cross point short is expected. Second one is the extra cross point suggests a faulty relationship at a convergence where no affiliation was expected. In perspective of their impact of the cross- point fault on the system functionality limit of the PLA, these faults are moreover named PLA faults.

1.4.11 Redundant Fault: A fault which is not changing the functionality of the circuit is known a redundant fault[120]. In a redundant fault the input and output relationship is not changing. So, it is difficult to identify the redundant fault by any test method. These types of faults are expelled by the circuits where the outputs are affected by fault inside. Multiple faults in a circuit may compensate the faulty behaviour of the outputs. Be that as it may, recognizable proof and expulsion of excess issues is a more unpredictable process. These faults can be observed in both combinational and sequential circuits. Generally, the issues in sequential circuits which any test related to sequential circuit testing, not able to detect the faulty behaviour of the circuit is called a redundant fault.

1.4.12 Memory Faults: Faults observed in memories are called as memory faults. Cell stuck at faults, coupling faults, address decoding faults are examples of memory faults[116]. Memories available in various structures and sizes. It is difficult to arrange fault model for each kind. For the most part by the basic structure of reading and writing of memory arrays, a memory can be all around depicted. Faults memories can be displayed, and algorithms to find faults in memories can be made by considering these read and write cycles. Memory arrays comprises of Analog and digital parts. Basic fault testing of memories may impractical. As a result of the

substantial number of cells and basic components. So functional testing model is conceivable as for memory testing.

Memories accessible in numerous structures and sizes. It is hard to configuration fault model for each type. Generally, by the common structure of reading and writing in to memory locations, a memory can be well described. Faults in memories can be modelled, and algorithms to discover faults in memories can be created by considering these read and write cycles. Fault can be characterized as the abstracted functional level of a defect in a digital system. Memory blocks may have analog circuit blocks and digital circuit blocks. Structural fault testing of memories may not possible as memories have large number of storage cells and other logic components. So functional testing model is possible with respect to memory testing. Functional modelling of memory faults consists of Stuck at faults, Transition faults, Bridging faults, Coupling faults, Data retention faults, Address decoding faults and Intermittent faults [27].

1.4.13 Stuck at faults (SAF): There are two types of stuck-at- faults. Stuck-at-zero (SA- 0) and Stuck-at-one (SA-1) fault. A SA-0 fault occurs when a cell is ‘stuck at logic-0' and does not change in response to any logical input pattern presented. Similarly, when a cell is SA-1 and does not change for any logical input pattern presented, that form of problem is referred to as a stuck at one fault [28].

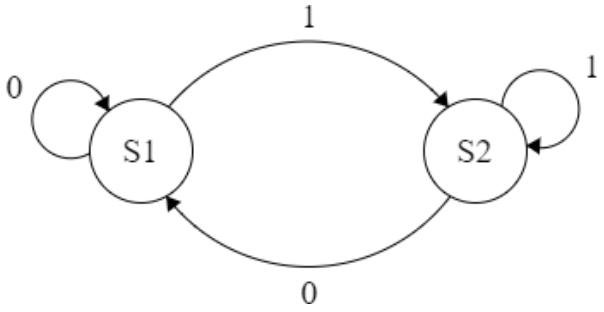


Figure 1.4(a) State Diagram of a Fault-free Circuit

Source: embedded.com

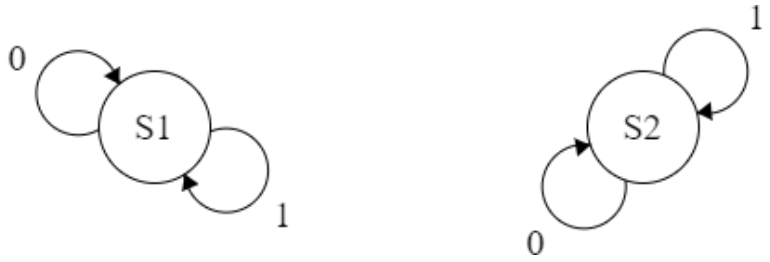
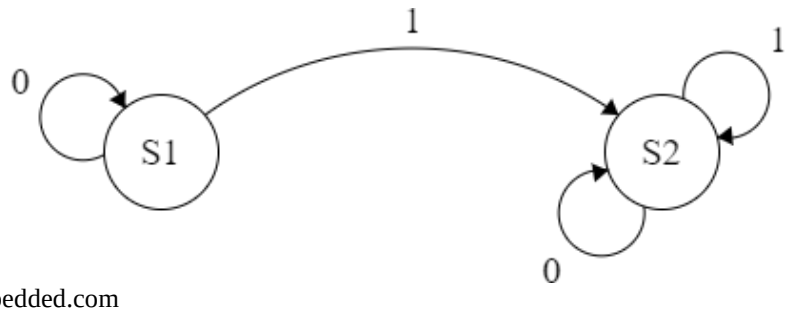


Figure 1.4(b) Stuck at Zero and Stuck at One Fault State Diagrams

Source: embedded.com

1.4.14 Transition fault (TF):A transition error occurs when a signal or line fails to toggle from ‘logic 1' to ‘logic 0' or from ‘logic 0' to ‘logic '1' [29]. In the case of memories, a cell fails to toggle. A rising transition error occurs when a cell fails to flip from logic '0' to logic '1'. A falling transition fault occurs when a cell fails to flip from logic '1' to logic '0'.



Source: embedded.com

Figure 1.5 State Diagram of Transition Fault

1.4.15 Coupling Faults (CF): When a cell is purposely transformed, the other cell changes, which is not supposed to alter. This is referred to as a coupling fault[22]. In general, when a cell is written with a logic value ranging from 0 to 1 or 1 to 0, the next cell is affected and its logic value changes. Different sorts of coupling defects have been identified in memories. A cell can only store a logic value if it contains a specific logic value, either 0 or 1. State coupling faults are the name given to this sort of fault. When a cell switches its state, the adjacent cell's value changes, which is known as an idempotent coupling defect.

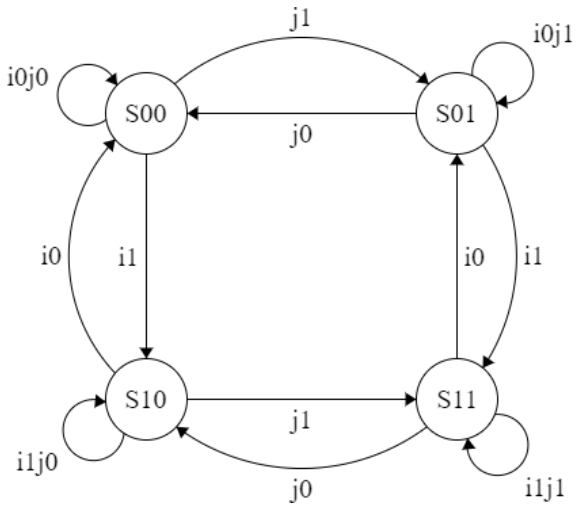


Figure 1.6(a) State Diagram of Two Cell Transitions, Fault-free

Source: embedded.com

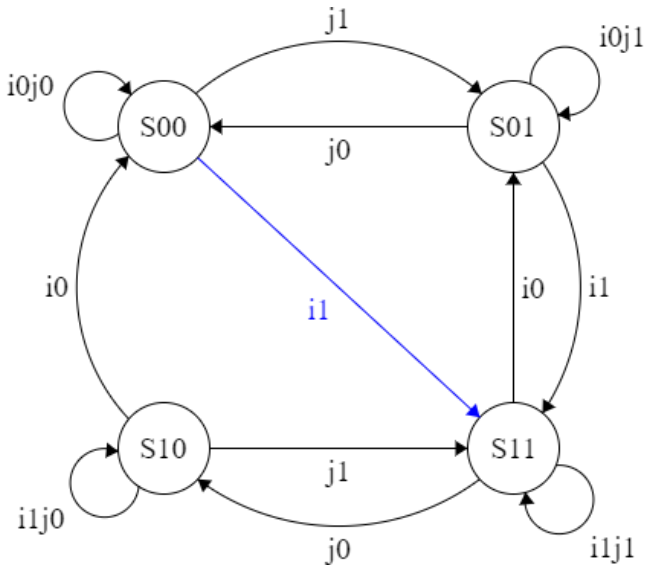


Figure 1.6(b) State Diagram of Two Cells with Coupling Fault

1.4.16 Bridging Faults (BF): These faults occur because of a short between some memory cells[36-45]. In bridging fault one cell alters another cell, there is no question of one cell dominating other. If there are two cells, namely cell_A and cell_B are affected in bridging fault. Either cell_A may influence cell_B or cell_B may

influence cell_A. Bridging faults are of two types. AND bridging fault and OR bridging fault. And bridging fault involved in two cells, namely cell_A and cell_B, when one cell is at logic zero then it pulls other cell also to logic zero.

When cell_A at logic 0 and cell_B supposed to be a logic 1 but cell_A pulls cell_B also at logic 0. In the same way cell_B when it is at logic zero it doesn't allow cell_A to go logic 1. This type of bridging fault is called AND bridging fault. OR bridging fault is when any one of the two cells at logic 1 then it pulls other cell also to logic 1. This type of fault is called OR bridging fault. When cell_A is at logic 1 and cell_B if supposed to be at logic 0, cell_A pulls cell_B also logic 1 level. In the same way when cell_B is at logic 1 and cell_A if supposed to be at logic 0, then cell_B pulls, cell_A to logic 1 level. This type of fault is called as OR bridging fault.

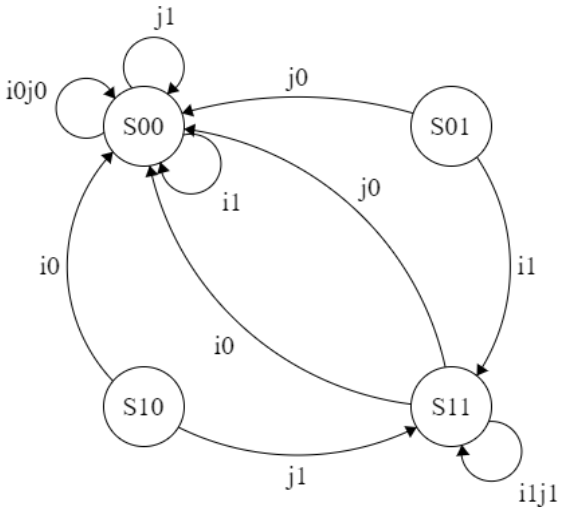
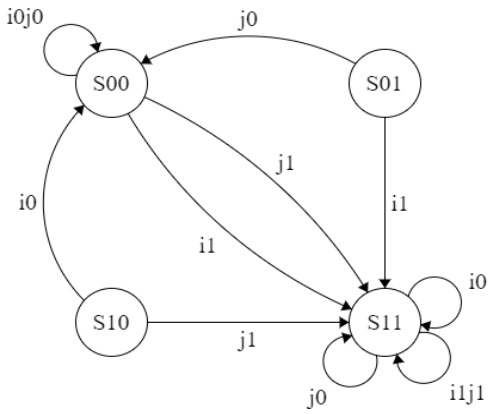


Figure 1.7(a) State Diagram of AND Bridging Fault

Source: embedded.com



Source: embedded.com

Figure 1.7(b) State Diagram of OR Bridging Fault

1.4.17 Address Decoder Faults: Address decoder failures are caused by incorrect address decoding in the address decoder. In general, four types of address decoder defects have been reported in memory[86]. No address can ever be used to access a cell array. The second condition occurs when no memory array is accessed with an address. Third, a memory array can have numerous addresses. Finally, numerous addresses are attempting to access the same cell array.

1.4.18 Neighborhood Pattern Sensitive Faults (NPSF): NPSF faults are due to influence of neighbouring cells of a call in an array. Because memory is closely coupled neighbouring cells affect the behaviour of a cell[100]. The pattern that is stored in the affecting cells alters the behaviour of affected cell. This is called Neighbouring Pattern Sensitive Fault. NPSF faults can be broadly classified in to two ways. One is Active NPSF and the other is Passive NPSF. Active NPSF is when a change in the any four adjacent cells complements the value in the affected cell. Passive fault is when a cell is surrounded by four neighbouring cells and all the four cells at logic 0 level then the affected cell stuck at logic zero level. In the same way when a cell at logic 1 surrounded by four cells at logic 1 then the affected cell stuck at logic1. This type of fault is called passive NPSF.

1.4.19 Data Retention faults: When a stored logic value in a memory array is not updated during a specific time interval, the cell that is unable to preserve its logic value is referred to as a Data retention error[115]. After a time lapse, the cell is unable to recover its previously stored cell content. These types of defects are common in SRAMs.

1.5 Concept of Fault Primitives

The fault primitive space is defined by Hamdioui [30]. An operation sequence that can be done in a memory cell. The FP is a precise, concise mathematical description that explains the operation done in the memory cell, the cell contents after sensitization, and the sense amplifier response. Using the concept of fault primitives, one may deduce all forms of faulty behavior, and it offers a framework for all errors in the memory cell. The two basic components of a fault model are: 1) A list of memory operations completed 2) A collection of corresponding variations in observed behavior from expected behavior. An operation sequence is a collection of memory cell actions. A sensitizing operation sequence (S) is one that produces a discrepancy between expected and observed behavior. Faulty behavior (F) is observable behavior that differs from the predicted one. FP that include a single cell is indicated as <S/F/R >, where 'S' indicates the sequence of sensitization operations carried out in the memory cell, 'F' indicates the response of faulty cell, and If the sensitization operation performed on the cell is a read operation, 'R' denotes the logical value that appears at the sense amplifier's output. '-' represents any operation. 'R∈{0,1,?,-}'. Here, '?' represents random or undefined logical value. A '-' in R represents that the output is not applicable in that case and the sensitizing operation does not contain a read operation.

Two cell functional fault models consist of fault primitives sensitized by performing utmost one operation while considering the effects of two cells have on each other. '<Sa,Sv/F/R> or <Sa,Sv/F/R>' the subscripts a,v indicates the fault primitive of two cells. The sensitization operation or state of the aggressor cell is denoted by Sa, while the sensitization operation or state of the victim cell is denoted by Sv. The set of Si is given as 'Si Ɛ {0, 1, X, w0, w1, w↑, w↓, r0, r1, 0w1, 1w0}', where I ∈ a, v and X represents the don't care value X∈{0,1}.

1.5.1 Single cell static FFMs

Single cell errors arises in the same cell,. This section summarizes many forms of single cell fault models and fault notations. Table1.1 summarizes the behavior of various forms of single cell static faults, as well as their acronyms and FPs.

Table1.1Single cell static Functional Fault Models

Type of fault	FPM	FPs
Single cell static faults	Stuck-at-Fault (SAF)	< □/1/- >, < □/0/- >
	Stuck-Open-Fault(SOF)	{< 0w1/0/- >, < 1w0/1/- >, < rx /x/?>}
	Transition Fault(TF)	< 0w1/0/- >, < 1w0/1/- >
	Read-Destructive-Fault (RDF)	< r0/↑/1 >, < r1/↓/0 >
	Deceptive-Read-Destructive-Fault (DRDF)	< r0/↑/0 >, < r1/↓/1 >
	Incorrect-Read-Fault (IRF)	< r0/0/1 >, < r1/1/0 >
	Random-Read-Fault (RRF)	< r0/0/? >, < r1/1/? >
	Undefined-State-Fault (USF)	< w0/?/- >, < w1/?/- >, < r0/?/? >, < r1/ ? /?>

1.5.2 Double cell static FFMs

In a coupled fault, two or more cells are coupled with each other. When two cells are involved, it is known as double cell fault. A two-cell coupling fault manifests itself if the contents of the victim cell is changed by a transition in the aggressor cell. LetC2, C3 and C4 be the neighboring cells of cell C1 (Figure1.8).

A coupling fault occurs mainly due to shorts in interconnections between adjacent cells. Shorts between neighboring cells can occur row wise, column wise, or diagonal. For example, between cells C1 and C3 in Figure1.8.

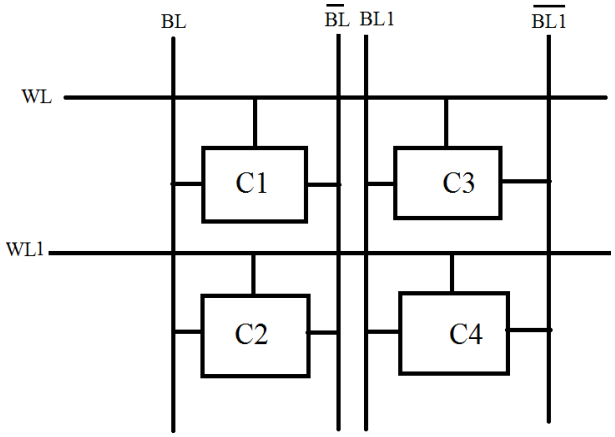


Fig 1.8. Coupled cells

- Static double cell errors are those that involve two cell actions. This section describes the classification of static coupled FFMs. Table 1.2 shows the behavior of numerous forms of double cell static faults, as well as their acronyms and FPs.

Table 1.2 Double cell static FFM

FFM	FPs
State-Coupling-Fault (CFst)	< 0;0/1/- >, < 1; 0/1/- >, < 0;1/0/- >, <1;1/0/- >
Disturb-Coupling-Fault (CFds)	< 0w1;0/↑/- >,< 1w0;0/↑/- > ,< 0w0;0/↑/- >,< 1w1;0/↑/- > < 0w1;1/↓/- >,< 1w0;1/↓/->,< 0w0;1/↓/- >,< 1w1;1/↓/->, < r0;0/↑/- >, < r1;0/↑/- >, < r0;1/↓/- >, < r1;1/↓/->
Read-Disturb-Coupling-Fault(CFrd)	<0;r0/↑/1>, <1;r0/↑/1>, <0;r1/↓/0>,<1;r1/↓/0>
Deceptive-Read-Disturb-Coupling-Fault (CFdrd)	<0;r0/↑/0>, <1;r0/↑/0>, <0;r1/↓/1>,<1;r1/↓/1>
Incorrect-Read-Disturb-Coupling-Fault (CFir)	<0;r0/0/1>, <1;r0/0/1>, <0;r1/1/0>,<1;r1/1/0>
Random-Read-Coupling-Fault (CFrr)	<0;r0/0/?>,<1;r0/0/?>, <0;r1/1/?>,<1;r1/1/?>
Transition-Coupling-Fault(CFtr)	<0;0w1/0/->, <1;0w1/0/->,<0;1w0/1/->, <1;1w0/1/->

1.6 Motivation:

From the literature review, the following points are noticed for testing of embedded memory. There are numerous testing methodologies available, but there are just a few strategies known for memory testing. All of these memory testing approaches are based on memory read and write activities. March memory test designs are popular memory test approaches. There are numerous March-based algorithms for testing SRAMs. Memory testing is becoming more difficult as technology continues to diminish. As technology advances, more and more faults develop in memories. Traditional memory test procedures may not meet today's low power test circuit's needs. Low power memory test approaches are required to overcome all of these disturbances. Memory is tested using built-in Self-Test procedures. March algorithms are commonly used to test memory, such as Static Random-Access Memory (SRAM). There are various March test strategies accessible to test memories. Each test approach has a distinct feature for testing various faults detected in memories at various levels of testing. March algorithms used a low power strategy to generate low power consumption Memory Built in Test structures.

Design engineers treat the Circuit Under Test (CUT) as a black box, as they see external input signals that can be controlled and the output signals that can be observed. As the complexity of the digital circuit increasing the testability of the circuit decreases. The design of a good Design for Testability (DFT) circuit is difficult as the circuit complexity increases. When a circuit became more controllable and more observable then it is more testable circuit. Test methods to enhance controllability and observability leading to enhance Testability of the digital circuit. By adding extra test hardware Controllability and observability can be improved. This extra hardware results extra control pins and output lines. More control pins and output lines results circuit complexity, high power consumption, high delay, less reliability and high cost. Effective test structures are needed to overcome all these trade-offs.

Built in Self-Test (BIST) is a hardware logic constructed along with the circuit to test itself. BIST is for generating test data to test Circuit under Test and observe the response and compares the output with expected results. BIST circuitry designed for testing memories is called Memory Built in Self-Test (MBIST). Since almost all the

systems uses memories, system behaviour completely depends on memories. It is mandatory to construct effective test methods for memories. Fault analysis and fault location are of prime significance in semiconductor Random-Access Memories (RAM) regarding the expanding memory utilization and commanding segment of installed memories in recent embedded processors. Manufacturing faults ought to be recognized, analysed and situated for additionally repair keeping in mind the end goal to enhance the item quality, dependability and yield.

BIST is the ability of a circuit or a chip or a board or system to test itself [16]. Built in Self-Test speaks to a collection of the ideas of Built in Test and Self- Test. The more equivalent term Built in Test Equipment (BITE) alludes to the equipment and additionally programming consolidated into a unit to give design for testability or BIST capacity. Now a day's low power advanced circuit configuration has risen as a key subject in hardware industry. Till now the VLSI industry builds more focus on chip area, product cost and performance. Now the VLSI architects focusing more on portable digital circuits.

In conclusion, the existing test methods are not considering all aspects of SARM parameters. This has motivated in developing a test method for single cell faults based on parasitic extraction method considering all aspects of SRAM parameters. The proposed parasitic extraction test method uses defect induced layout, which is extracted from known electrical model.

Due to complexity in representing coupling faults, research was done at low rate compared to single cell faults. This problem directed the researcher to extend towards analysis of coupling faults too. In this process, a fault model dictionary was developed by the researcher for detecting existing, detectable and undetectable single cell and multi cell SRAM faults, in 45nm, 32nm and 7nm technologies.

1.7 Objectives and Scope of the Work

It is necessary to thoroughly evaluate memories using effective algorithms and realistic defect models. By developing a fault model dictionary at three technology levels and employing the parasitic extraction method, the investigation tackles the issue of test time complexity in the development identification of a variety of faults in

the new technological era while taking parasitic memory effects into consideration.

With the continuous downscaling of silicon-based technologies, it is felt that there is a growing need for mechanisms to detect the manufacturing faults. Bridging/short and open faults are very common manufacturing faults. The researcher analyzed the open and short faults for single cell SRAM. The proposed parasitic extraction method has been implemented using MicroWind tool for three technologies 45nm, 32nm and 7nm technology.

With technological advancement, AI and Machine Learning are playing a very key role in every industry. Machine learning (ML) has benefited the VLSI industry by optimizing the use of EDA tools, resulting in reduced design times and manufacturing costs. Machine learning in VLSI design helps EDA tools find the best solution for use case scenarios and minimizes production loss by recognizing chip faults. Therefore the researcher used the machine learning algorithms to predict the faults.

With the continuous downscaling of silicon-based technologies, it is felt that there is a growing need for mechanisms to detect the faults. In this scenario, an outcome based analysis carried out using parasitic extraction method will be necessary to guarantee an accurate testing method. The proposed methods are envisaging the future needs.

1.7.1 Objectives of the proposed work

1. Analyze faults due to open defects in Embedded SRAM cell for submicron to deep submicron technologies
2. Design a parasitic R and C extraction method for short/bridge defects in SRAM cell using nano meter technologies.
3. Develop a fault detection method for open and Short defects in multi cell SRAM architecture
4. Design Novel Fault detection and test methods using Machine Learning Algorithms in embedded SRAM architecture

1.7.2 Tools Used for the proposed work

1. Microwind Tool (3.9 licensed Version)
2. Multiple Linear Regression
3. Decision Tree algorithm

1.7.3 Organization of the thesis

The significance of embedded SRAM testing when they are placed in SoCs and brief details on different faults and fault models of eSRAM with their implementation methods were discussed in Chapter 1. The criterion is to reduce the test time in addition to the complete fault coverage.

The efforts put by several researchers to develop algorithms for detecting faults, and the different fault models develop by them are discussed in chapter 2.

The necessity of identifying faults by considering parasitic memory effect and then the details of systematic procedure adopted to extract the parasitic values using 45nm, 32nm and 7nm technologies is discussed in chapter 3. Different experiments are carried out to arrive at the fault model dictionary for single SRAM cell and multi cell SRAM that can be used for testing are presented.

The importance of the Machine Learning in VLSI and different type of machine learning algorithms used to predict the Parasitic R and C values and faults in SRAM discussed in chapter 4.

Finally, in Chapter 5, the summary of the experimental results obtained and the concluding remarks are presented.

CHAPTER 2

LITERATURE REVIEW

The recent research on testing techniques for detecting resistive open defects and an impact of the resistive open defects on circuit operation has been summarized. The timing dependent effects described from the resistive opens and these are tested by using the delay test. The delay fault test is using for detecting the open defects which creates more delay than the expected delay. Due to these delay faults cause a malfunction of the Integrated Circuits. By using this delay fault test, also detects weak open defects when it causes longer delay in the longest path in fault free integrated circuits. Moreover, the recent research on design methods of SRAM circuits has been summarized.

The ROF is one of the major fault which present during fabrication of ICs. The ICs are fabricated by using several number of transistors and all are interconnected with each other. The presence of ROFs in ICs degrades the conductivity between two nodes of inter connections. Due to degradation in conductivity causes delay fault. Based on value of ROFs these are classified as gross delay fault, small delay fault and undetectable delay fault. The presence of ROFs in SRAMs changes the operation.

The testing of nanometric low power digital integrated circuits primarily rely on the types of fault explained by Khursheed [2010] [31] which is meant to mimic the physical behavior of defects while considering into account all the physical details connected with the behavior of a defects at the device/system level. The test patterns generated and the stimulation of defect behavior is made by using fault models. There are different physical defects, viz. Resistive Short, Resistive Open, Transmission Gate Open, Gate Oxide Short, threshold voltage shifts, diminished drive strength etc. Therefore the higher level of abstraction can be captured using fault models.

The Resistive Bridges is also represents a major section of defect in deep sub- micron CMOS. Moreover, received an increased attention with regard to modeling, test generation as well as diagnosis.

The process variation and physical defects in ICs causes the Delay Faults. The ROFs

and resistive bridge faults described by Velaga [2013][32] produces excessive circuit delays as well as violate circuit timings. At the inputs of latches or flipflops the data setup time violations are caused by delay faults. These faults make manufactured ICs to fail for operating at desired frequency of operations.

ROFs are also attracting significant research interest around the world in order to reduce testing costs while maintaining fault coverage ranges in the context of different VDD systems. According to several recent research investigations, full open defects can be checked using static testing methodologies in any VDD situation because they do not display VDD dependent detectability. On the contrary, resistive-open faults are significantly more observable, and their prevalence increases when several VDD settings are used in conjunction with various delay test procedures.

Dilillo *et al.,* [2004][33] harnessed the march test for detection of ROFs. The significance of ROFs has considerably increased in recent technologies due to presence of many inter connection in fabrication of ICs. In this work considered 6T-SRAM circuit and injected deferent valued ROFs inside this SRAM circuit. Tested the circuit performance of SRAM circuits after injecting ROFs. Based on value of ROF, these are classified into different types. These are 'Read Destructive Fault, Deceptive Read Destructive Fault, Dynamic Read Destructive Fault, and Incorrect Read Fault and Transition Fault'. In this work given that, 'Transition Fault' induced when ROF value is greater than4 0KΩ, 'Read Destructive Fault an Deceptive Read destructive Fault' induced when ROF value is greater than 14kΩ, 'Dynamic Read Destructive Fault' induced when ROF value is 140MΩ and 'Incorrect Read Fault' induced when ROF value is greater than2 00 KΩ. In this work proposed a unique march test for covering all these faults caused by ROFs. The drawback in this work was given about only fault presence during read operation, not explained about presence of ROFs during write operation.

Martins *et al.,* [2016] [34] analysed Negative Biased Temperature Instability (NBTI) which causes cells aging impact on the SRAMs if ROFs present. In this work also classified faults into different types based on ROFs value. These are 'Read Destructive Fault, Deceptive Read Destructive Fault, Dynamic Read Destructive Fault, Incorrect

Read Fault and Transition Fault'. In this compared the performance of 6T SRAM without and with ROFs. In this work mainly concentrated on testing performance of the 6T SRAM when weak ROFs present during fabrication. The presence of weak ROFs in 6T SRAM cell shows more impact on NBTI and causes memory cell aging. Difficult to detect weak ROFs in SRAM cell. But slowly it shows more impact on NBTI of SRAM cell. The major drawback in this work was not defined the weak ROF Value.

Arumí et al., [2015][35] analyzed an effect of the downstream parasitic capacitances on Struck Open Faults (SOFs) feature. It was also demonstrated on how these capacitances became more contributors of testing the escapes for speed testing, even greater than leakage currents. The proposed method was increased the detectability of the fault in 65 nm technology. The main disadvantage was that it maximizes the number of the test escapes when SOFs was affected in the parallel n-network.

Li *et al.,* [2003][36] developed an efficient variation aware delay faults simulation method for detecting the resistive bridge as well as resistive open defects. The proposed method was achieved approximately 53 times faster with ≤4.1 per cent error in accuracy for detecting resistive opens and approximately 40 times faster with ≤ 5.1 per cent error in accuracy for detecting resistive bridged efects. The disadvantage of this method was that its initial level the input gates increase the error and hence the speed is also reduced.

Azevedo *et al.,* [2014][37] analyzed an impact of the resistive opens on Thermally Assisted Switch (TAS) Magnitude Random Access Memory (MRAM) behaviour. In this work designed and simulated hypothetical 16 word TAS-MRAM structure for performing both read and write operations. The proposed method has a major advantage, it was reducing the selectivity problem. The problem in this method was that the TAS technique requires an additional current compared to the second magnetic field in the field induce magnetic switching MRAMs.

Yu *et al.*,[2014] [38] Developed attest method of the transient power supply current used for the ROFs and employ wavelet analysis for locating the fault. In this work considered one inverter chain circuit and injected resistor which present ROF and tested .The combination of both the transient current as well as wavelet methodology

in the digital circuit is used for testing resistive open defects effectively. Because average transient current method can't detect the position ROFs.

Renovell et al., [2006][39] was proposed a Specific Automatic Test Pattern Generation(ATPG) method for detecting ROF. In this work analysed the electrical behaviour of resistive open defects as a function of their unpredictable resistances. It also considered predictable resistance value as a node. In this specific ATPG method considered a given node with a ROF and generated a test vector propagated a rising/falling transition via longest path including the faulty path. The disadvantage in this method was that it doesn't specify the detection range of the ROF value.

Then the next ROF detection method, i.e. the stuck at fault and transition fault testing techniques used for detection and these techniques developed by Czutro*etal.*,[2008][40]. If the size of delay faults less than one clock cycle which are caused by resistive opens, categorized as small delay faults. A new simulator was implemented for the detection of these very small delay faults which are caused by the resistive opens. These small delays can't be detected in the normal testing methods. In some cases, these very small delay faults can also cause problems while circuit operation. The disadvantage in this method was that it detects only very small delay faults didn't explained about detection of medium and large delay faults.

Yang *et al.,* [2004][41] proposed a SAT based ATPG for detection of ROFs. In this fault model resistive open defect was detected through the longest sensitized path. In this work the quality this proposed test was also analysed by comparing this test with 3 other test sets. The 3 tests are single detection transition fault test sets, multiple detection transition fault test sets as well as traditional critical path test sets. The drawback in this proposed method was that it mentioned the defects injected in the circuit but not explained.

Tahoori [2002][42] utilized one a new method for testing of ROFs in FPGAs. In this method delay of the faulty path also increased many times more than delay of the fault free path. The proposed work has also taken a 3-stage inverter circuit as an example circuit and injected resistance for testing. The drawback in proposed method was that it did not explain up to which value the ROF can be detected.

Haron and Hamdioui [2012][43] Used Design For Testability (DFT) schemes for the detection of ROFs in Resistive Random Access Memories(RRAM). The new DFT schemes proposed for detection of resistive open fault in RRAMs were short-write time based DFT as well as Low-write voltage based DFT. These proposed DFT schemes used the resistive open fault injected circuit for testing. The drawback in proposed method was that it is not possible for detecting fault which is below 56kΩ value of ROF.

Montañés et al., [2002][44] applied yield evaluation monitor method for detecting weak open-line defects. They defined resistance ranges of various ROFs like Weak Open-[1MΩ-10MΩ] Strong Open-[10MΩ-1GΩ].Moreover, they have given utmost importance in the detection of the weak open defect because the strong open defect can cause a circuit for malfunctioning but weak open defect cause it to poor function. The drawback in the proposed method was that detection is done only during the fabrication stage.

Krishnan and Theepa [2015][45] developed scab Based BIST Techniques for detection of resistive open faults. In this classic BIST architecture resistive open faults detected by observing the performance violations for various Vdd values captured in digital signatures, which were significantly differ from the fault free signatures. The draw back in proposed method was that it did not explain about the range of fault coverage.

Maharana et al., [2016][46] proposed oscillation based BIST, Look Up Table (LUT) based low power testing techniques for testing low power Analog circuits. In this work fault coverage is identified by injecting fault and fault simulation. In this proposed work taken operational amplifier as an example circuit and injected resistor to present fault. The drawback of this proposed design was not given the fault coverage range of open faults detection.

6T SRAM and 7T SRAM were designed and simulated using three distinct technologies: one eighty nm, ninety nm, and forty five nm by Chhillar et al., [2013][47]. Simulating both 6-Transistor and 7-Transistor SRAMs yielded static, dynamic, average power, and stability measurements. The 7-Transistor SRAM improves write speed over the 6-Transistor SRAM while consuming more area. The

Read Static Noise Margin and Write Static Noise Margin of 7-Transistor SRAM were improved over 6-Transisitor SRAM. The drawback with this work was that the leakage current and latency were not explained.

Kumar et al. [2014][48] proposed a 6T SRAM using ninty nano meter and one eighy nan-meter technology. In this paper they explained how the read write operations will affect by scaling the transistor. They have compared proposed 6T SRAM cell's performance with a regular 6T SRAM cell. This work involves scalability in the design of a 6T SRAM. By scaling down the transistors the power dissipation can be reduced but this will effect on the stability of Read/Write operation. in their proposed work the main draw backs are more latency and high leakage current.

Shiva prakash and Suresh [2016][49] investigated SRAM performance using SNM, read margin, and write margin parameters in their paper. SNM is a critical factor influencing the stability of read and write operations. Because of the high speed of the 6T SRAM, the data retention voltage was computed. The suggested work had the disadvantage of just providing SNM analysis.

Munaf et al., [2017][50] proposed FinFET based 6T SRAM. The parameters of conventional and FinFET SRAM Cells were compared. In this work also designed traditional 6 transistors SRAM, ST based 6transistors SRAM and CNTFET based 6 transistors SRAM cells. The review report was prepared by designed and simulated the conventional SRAM, ST based SRAM, CNTFET based SRAM and FinFET based SRAM circuits. The drawback in proposed work was that it has the delay is more for write operation.

Deora and Shrivastava [2018][51] proposed This double gate FinFET based SRAM designed with 90 nm technology. The using double gate FinFET technique reduced the power consumed by SRAM cell. In this work measured the values of area, power and delay parameters.

Jennifer et al., [2018][52]compared the 6T SRAM & 7T SRAM with respect to the power dissipation. In proposed design simulation results shows the 7TSRAM has more power dissipation than 6TSRAM. The drawback in proposed work was that explained about only power analysis of 6T as well as 7T SRAM cells.

Parihar and Jangid [2015] [53] designed SRAM Cell by using 6 transistor, 7 transistor and 10 transistors with 45 nm technology. In this measured and compared the static and dynamic power of proposed SRAM cells. The proposed work mainly focused on the reducing of power dissipation during short circuit and also reducing of dynamic power. The10T SRAM has better read stability and write ability than 6T SRAM but 6T-SRAM has low power dissipation. The drawback in this design was has more delay.

M.Parvathi et al. [2015][54] proposed parasitic extraction method to test the SRAM cell for short defects from the layout diagram. In the proposed method they have extracted the parasitic Resistance and capacitance for fault free SRAM cell. These values are used to compare with extracted R, C values of faulty SRAM cell to detect the manufacturing defects.

Khare et al., [2008][55] proposed 6T SRAM for low power applications. This 6T SRAM designed with 35µm technology and simulated by using cadence tool. The operating voltage was given as 1.8 V. In this design the conventional bit line BL was divided into shorter bit lines. The short bit line has less capacitance compare to the long bit line. These bit lines reduced the power dissipation. The disadvantage in this design was that it has more delay for read and write operation.

Kumar and Chalil [2019] [56] proposed 90 nm 6T SRAM by using the FinFET Technology. In this design comparison of SNM and leakage current between MOSFET and FinFET transistors was given. In this work determined the stability of SRAM cell by using butterfly method of static noise margin analysis. In this design defined that the FinFET transistor has small leakage currents compare to the MOSFET transistors. At room temperature the leakage current measured as a 3.59E-10 and it is small than compare to the MOSFET based 6T-SRAM. This proposed 90nm 6T-SRAM has 43.894nA. The drawback in this design was has more power consumption.

Noor et al., [2019][57] designed 6TSRAM with 20nm technology Silicon on Insulator(SOI) junction less transistors. Simulated the proposed 20nm SOI junction less transistor based 6T SRAM by using the Sentaurus Device Editor module of Synopsys Sentaurus TCAD software. The Predictive Technology Model (PTM)

SPICE model generator was used for design of SOI junction less transistors. In this work improved the read stability and write ability of proposed SOI junction less based 20nm 6T SRAM more than 20 percent. The drawback in proposed design was reduced the read and write SNM. Premavathi *et al.,* [2016][58] implemented 6 transistors SRAM with separate read access path and write access path by using the FinFET technology. The proposed SRAM circuit was designed by combining the advantages of both the conventional 5T and 8T SRAM cells. This design enhanced write ability and readst ability. The main drawback in proposed work was that the delay of the write operation increased.

Thomas *et al.*,[2009][59]implemented 45nm SRAM cell in monolithic 3D IC technology by using TCAD extraction. In their work, they have designed traditional 6-Tansistor SRAM and performance compared with proposed design. The static noise margin improved 10 per cent and static power consumption reduced 12 per cent. The disadvantage in proposed design was that it has not given the delay of read and write operation.

Prachi and Neetu [2013][60] focused on write and read operations of 6 transistors SRAM peripheral circuitry such as a column multiplexer, row address decoder, sense amplifier which were the main building blocks of SRAM cell. The drawback in proposed work was that delay increased with number of sections. Kumar and Kingra[2016][61] proposed 90nm 6transistors SRAM for enhancing stability of read and ability of write operations. The stability of read and ability of write of the SRAM cell depends on SNM. In this work calculated the both read and write SNM of proposed design. Here also used butterfly curve method for the calculation of SNM. The drawback in proposed design was that the supply voltage Vdd given is high.

Tripathi *et al.*,[2018*b*][62] proposed multi thread SRAM cell for improving stability and low leakage currents. The proposed SRAM cell simulations were done by using the Cadence Virtuoso tool with UMC 55nm technology. This multi threshold CMOS SRAM cell designed by using additional two transistors along with 6T SRAM cell. In this work, traditional 6T SRAM performance was compared with the suggested

architecture. The drawback in proposed work was that the transistor count was increased.

Kumar and Tomar [2019][63] proposed sub-threshold SRAM for achieving stabilized read and write operations. The sub-threshold voltage is defined as the reducing of supply voltage VDD below threshold voltage (Vth). In this work body biased technique was proposed for improving the static current noise margin. The drawback of this work was not explained about other parameters.

Saun and Kumar [2019][64] SRAM cell was designed and analyzed at several technologies by using the Predictive Technology model(PTM) for reducing t he power dissipation and maintaining the stability. The drawback in the proposed work was that it has more delay.

Kassa and Nema [2019][65] designed Static RAM cell by using Quantum dot Cellular Automata technique. This design circuit simulated by using QCA Pro and QCA Designer 2.0.3tool. This proposed design reduced energy power dissipation of SRAM cell. The drawback in this proposed work was only given energy power dissipation analysis not explained about leakage currents and delay.

Fan et al., [2012][66] designed FinFET based SRAM circuits mainly effected by single charge trap induced random telegraph noises. The Random Telegraph Noise (RTN) of FinFET based transistor generated by trapping or de-trapping of carriers at interface trap. In this work evaluated the influence of single charge trap which induced RTN. The drawback in this work was concentrated on only RTN.

Gupta and Anis [2009][67] proposed a statistical method for designing of SRAM cell to ensure a high memory yield. Variability in the dimensions of transistors caused by sub wave length lithography and proximity. Due to this variability causes intrinsic Vth variations. In this work mainly concentrated on controlling of these intrinsic Vth variations and stabilizing of 6TSRAM operations.

Kranti *et al.*,[2010][68] proposed a unction less SRAM cell. The proposed unction less memory cell designed with nano metre technology and performance compared with traditional 6TSRAM. In this work improved SNM of proposed junction less 6TSRAM. In this design not explained about other operational parameters of

6TSRAM cell.

Deng and Houston [2003][69] proposed SRAM design first transistor has a control electrode current and back gate/body connection electrically connected to the control electrode. The main disadvantage was it has more area.

The parasitic resistance and capacitance (RC) effects of a single- FinFET on logic CMOS devices were examined by Bo-Rong Huang et al.2017 [70]. In addition, the effects of dummy patterns and multifin structures are thoroughly investigated and modeled. In this study, the static and dynamic performance characteristics of 6T-SRAM cells are thoroughly analyzed as an example of parasitic RC effects by combining parasitic resistance and capacitance derived through measurement and simulation.

Jadav et al., [2012][71] proposed the method for ultra-low power operations, by using adiabatic technology. The ac power supply used in this a diabatic technique. In this proposed SRAM cell a diabatic switching achieved by kept potential across the switching devices are very small. This a diabatic technique in proposed 6TSRAM reduced average power dissipation up to 75% than compare to the traditional 6T SRAM design. And also improved the SNM. The disadvantage of this proposed 6T-SRAM design was that the leakage current and delay were not explained.

Bhaskar [2017][72] proposed Multi Threshold CMOS(MTCMOS) technique. In this MTCMOS technique used transistor with multiple threshold voltages for reducing delay and power. Low threshold voltage Vt transistors switch faster but has more static power. Similarly high threshold voltage Vt transistors switch slowly but has low static power consumption. This proposed MTCMOS based 6T SRAM design has better performance than compare to traditional 6T SRAM design. The draw back in this design was has higher power consumption and delay.

Bharti et al., [2019][73] proposed Floating Gate MOS based 6T SRAM design for ultralow power operations. This proposed 6 transistors SRAM designed with 90 nm technology and simulated by using Cadence tool. The FGMOS is an approach to deliberate a minimal power design which involved few transistors to maintain compatibility, flexibility and large tune ability. In this design at gate terminal the

FGMOS has infinite input impedance. The advantages of FGMOS than compare to MOSFET are small size and low power dissipation. The proposed 90 nm FGMOS base 6T SRAM cell has better performance than compare to traditional 6T SRAM design. The drawback in this FGMOS based 6T SRAM design was has high delay.

Bikki et al., [2019][74] proposed low power SRAM cell design with leakage control techniques. The leakage current depended on many factors such as supply voltage, dimensions and process parameters. The subthreshold leakage is major source of leakage current because of low threshold voltage used in nanometre CMOS technology. To reduce leakage currents in this proposed 6T SRAM design used power gating and body biasing techniques. In this work the proposed design has better performance. The limitation of proposed design was high power dissipation.

Saxena and Mishra [2016][75] proposed SRAM cell for reducing leakage power because more than 40 percent of power wasted due to leakage through the transistors in memory cell. In this work also designed traditional 6T SRAM and compared the performance with proposed SRAM design. The proposed 7 transistor SRAM design has better performance than compare to traditional 6T SRAM design. The conventional 6TSRAM design suffered from external noise margin owing to direct path through bit line to their storage node. In this proposed 7 transistor SRAM design has separate mechanism to perform read and write operations. The drawback in the proposed design was described only analysis of leakage power and leakage current.

Narah and Nath[2018][76] designed 6 transistors, 7 transistors, 8 transistors and 9 transistors SRAM with 45 nm , 65nm and 90 nm technologies by using the CADENCE TOOL. In this work measured the static power dissipation, leakage current, SNM and delay of 6 transistors, 7 transistors, 8 transistors and 9 transistors SRAM cells. Based on these parameters evaluated the performance of all the SRAM cells. For reducing power consumption performed scaling but scaling affected the other parameters of SRAM. The power consumption reduced for 45nm 6T-SRAM but stability decreased when compared with higher nano-meter technology SRAM.

Joel and Gnana [2017][77] proposed 7 transistors SRAM by using power gating techniques. The power gating techniques used for reducing power dissipation in

SRAM cell by disable current to blocks which were not in use. The power gating techniques are ground gating, supply gating and combined gating. The 7T SRAM cell at 45 nm technology shows far better power and temperature stability than traditional 6TSRAM design.

Sharif et al., [2018][78] proposed Dual Vt 7 transistors SRAM. In proposed SRAM used small size dual threshold voltage transistors in the cross coupled inverters for reducing the leakage power dissipation without affecting any degradation in speed of the read operation. The drawback in this design was not explained about the delay of read and write operation.

Ansari et al.,[2015][79] proposed a method for improving read and write margins. This 7 transistors SRAM designed with 20nm FinFET technology. In this work proposed 7 transistors SRAM design with differential write operation and single ended read operation by working in the near threshold voltage region. In this work also designed conventional 6T SRAM cell 20nm FinFET technology and performance compared with 7T SRAM cell. In this work used an extra access transistor for enhancing read operation static noise margins, write margins and reducing write time. And also reduced the leakage power by using an extra access transistor. In this work measured read margin, write margin, leakage power dissipation and delay of both proposed 7T SRAM design and conventional 6T SRAM design. Based on these parameters defined that the proposed 7 transistors SRAM design has better performance than compare to traditional 6T SRAM design. The drawback in this design was not clearly explained the designing procedure of 20nm FinFET 7TSRAM.

Liu et al., [2017][80] proposed SRAM cell design by using the Tunnel Field Effect Transistor (TFET) technology for ultra-low voltage applications. In this work tensile-strained Ge/InGaAs TFET-based 7TSRAM architecture was designed and simulated. The MOSFET has a limitation that it doesn't support the ultra-low voltage operations but TFET permits the ultra-low voltage operations. So, for ultra-low voltage applications proposed to design SRAM cell by using TFETs. In this work measured read static noise margin, write static noise margin, delay parameters and evaluated the performance of proposed 7T SRAM cell statically and dynamically. In this work also

designed traditional 6TSRAM design with 45nm CMOS technology and compared the performance with proposed TFET based 7T SRAM cell. The investigated outward access transistors provide successful read and write capability. The drawback in this proposed design was increased read delay times.

Sharif et al., [2017][81] proposed a new 7T SRAM design for improving SNM. The proposed 7T SRAM cell designed with 45 nm PTM models and simulated by using LT spice tool. In this work by controlling of drain induced barrier lowering improved the static noise margin and better read operation. In this work also designed conventional 6T and 7T SRAM cell and compared performance of these circuits with proposed 7T SRAM cell. The proposed design improved not only stability of read operation also improved over writing new data into memory cell. The main drawback in proposed design was its more delay than the conventional design.

Takashima et al., [2018][82] proposed 7T SRAM cell for improving stability of write operation. In this work capacitive coupling used in 7T SRAM for data written into memory. This capacitive coupling enhanced the stability of write operation in 7TSRAM. The suggested capacitive coupling-based 7T-SRAM cell improves the standard 6T SRAM in write stability.

Kumar and Tripathi [2020][83] proposed 7T SRAM cell for reducing leakage power and current. The proposed 7T SRAM designed with the 18nm FinFET technology. In this work used self-controllable voltage technique for reducing leakage current and power.The FinFET based 7T SRAM cell was faster and reduce the power consumption than compare to the MOSFET based 7T SRAM cell. But its drawback was has more leakage current.

Mishra et al., [2020][84] proposed a low power 7T SRAM cell design by using supply feedback technique. This proposed 7T SRAM cell designed and simulated by using Cadence Virtuoso tool. For improving write ability and reducing static power dissipation proposed this 7TSRAM design by using supply feedback transistor. And read stability increased by isolating bit_line nodes. The proposed 7T SRAM design has better performance than compare to than compare to the 6T and 8T SRAM cells. The drawback of this design was not described about leakage current.

Luigi Dilillo et al., in [85-87], large value defects in VDSM new nanometer technologies. The works explain behavior of dynamic Read Destructive Fault (dRDF) and propose read equivalent stress (RES) as a new test approach for sensitization of dynamic faults for detection.

Dilillo Luigi et al., in [88, 89], demonstrate that the pre-charge activities for all columns bit lines are undesirable when March test elements operations in progress. This reduces the unnecessary stress and minimizes power consumption in operations of March test elements processes. The proposed work is implemented by designing modified pre-charge activity controller logic circuit.

A. Ney et al., in [90], during processes of operations of test algorithm element uses information's of large band. These large information's gives almost complete information about location fault and defect in SRAM's core cell. Also this signature based approach is useful to distinguish dynamic fault and static fault in SRAM's core cell array and other memory components. This approach improves the test performance memory components without increasing the test complexity and modification in March test algorithm as compared other existing methodology.

Alberto Bosio et al., in [91], VDSM technology introduces new and high resistive open defect which creates new faulty behavior known as dynamic fault in SRAMs. So new test solutions with large coverage range, low complexity and low power consumption are necessarily required. According to these requirements work proposes a new March BDN test algorithm.

Luigi Dilillo et al., in [92], explain Complex Read Faults (CRFs) in SRAM. Study shows various error factors create fault during process of saved data read operation in very dense or nanoscale channel semiconductor SRAM memories core cell array. All the SRAM's parts during 'read' operation have been studied to understand real cause of fault. New fault model requirement has been given to explain this fault. Authors explain that various causes of 'read' operation failure are independent. The cumulative or summing effect of these causes creates or provokes Complex Read Faults (CRFs) in nanometer technology. In this work authors show that the steps of testing to increase the fault coverage range to detect this new Complex Read failure with all the various

‘read’ faults. Authors proposed March CRF as a very low complexity test algorithm it covers all the realistic Complex Read Faults effectively.

R. Alves Fonseca et al., in [93], L. Dilillo et al., in [94] and M. T. Mohammad at et al., in [95], present a study of bridging effect of defects which has resistance electrical equivalent. This bridge formation of resistive open defects generates dynamic and static faults not only in defective core cell but also in an adjacent defect free SRAM’s core cell.

Alexandre Ney et al., in [96-98], present analysis on 65 nm channel devices behavior, defects, and faults and propose March test algorithm for solution. Shows sense amplifier and write driver faulty behavior due to some defects which has resistive electrical equivalent. These faults create error in write driver and sense amplifier operation in SRAM’s core cell array. There are two types of fault cases. Due to type-1 case problem in any read operation and in type-2 case problem in particular read or write operation.

P. Rech et al., in [99], N. Mahatme et al., in [100] and G. Tsiligiannis et al., in [101], present exhaustive stressing factor and define device error rate evaluates the changes and faults due to exposed SRAMs in neutron radiation. Bosio A. et al., in [102], present the challenges of time consumption and power dissipation during testing in SRAM’s core cell array of VDSM technologies. Introduce new test architecture [103-107] for latest technology’s very high density SRAMs dynamic faults which sensitized by RES or read operations. Presents ‘RES’ sensitization for dynamic faults is very high efficiency then read operation.

Naik and Kuwelkar [2017][108] proposed 8TSRAM cell for optimizing power and delay. The proposed 8T-SRAM cell designed with 45-nm technology and simulated by with Cadence Virtuoso tool. In this work optimized and stable cell structure was designed by changing the W/L ratio of MOSFET transistors. The stability of proposed 8TSRAM cell found by doing SNM analysis with butterfly curve. The disadvantage of using butterfly curve for measuring stability was an automatic inline testers were unable to measure SNM. In this work for measuring SNM used N curve instead of butterfly curve. In this work measured SNM, power dissipation, read delay and write

delay and based on these parameters obtained that the proposed 8T SRAM cell has better performance than compare to the conventional 6T SRAM cell. The proposed work was not explained about the leakage current.

Raikwal et al., [2017][109] proposed 8T SRAM cell for enhancing read stability. The proposed8T SRAM cell designed with 180 nm technology by using the S-Edit as well as the net list and simulation was done by using the T- spice. In this work enhanced the write ability and read stability of proposed 8T SRAM cell by controlling of read, write operations with separate word lines. In this work improved the read static noise margin and reduced the power consumption of proposed design than compare to the conventional 6TSRAM cell. The drawback in proposed design was not explained about the write static noise margin.

Kolsoom *et al.*,[2015][110] for improving read and write stability then compare to the conventional design proposed a 7T SRAM design. In this design for improving read stability isolated read path from storing node. For improving write ability virtual ground one of the inverter used. This proposed 7T SRAM designed by using 90nm technology and simulated In HSPICE tool. The drawback in this was has high leakage current.

Yang [2010][111] proposed a new SRAM design using Bit line Charge Recycling. This CR SRAM reduced the read and write powers by recycling the charge in bit lines. This CR SRAM implemented in 0.35μm CMOS process. The drawback of proposed design was has high power consumption. Kahng [112] examines machine learning opportunities with a focus on IC physical implementation. (1) is an example of an application. Through correlation techniques, superfluous design and modeling margins are removed.(2) Accelerating design convergence using predictors of downstream flow outcomes that take into account both tools and design cases. (3)Additional benefits include optimizing the use of design resource licensing and available schedule.

Brett Shook and colleagues [113] developed a revolutionary machine learning-based parasitic estimation (MLParest) method for pre-layout customized circuit design. It decreases the average inaccuracy between pre-layout and post-layout circuit

simulation from 37% to 8% for a wide variety of analog circuit metrics. As a result, MLParest can reduce the number of cycles required between the pre-layout and post-layout process of design dramatically. This work's key contributions are a machine learning-based parasitic estimation method and a model training framework with a single button that is scalable across diverse technological nodes.

S.K.Samal et al. 2016[114] demonstrated that industry On Chip Variation (AOCV) tables cannot be directly applied to 3D paths spanning many dies. They created a new machine learning-based model and approach for estimating the diversity of logic routes in 3D architectures. Key parameters derived from an existing GDSII 3D IC design and sign off simulation database are used in their model.

Princy 2019 [115] Wavelet-based transient supply current test with modified March sequence exploiting Read Equivalent Stress (RES) for DRF detection was implemented as a unique technique to SRAM testing. The proposed test improves the detection of Data Retention problems. It is entirely dependent on software implementation.

Table 2.1 Comparison of Literature Review

Title	Year of Publication	Indexing of journal (Scopus/SCI index etc.)	Main findings or conclusion relevant to proposed researchwork
A New Test Algorithm and Fault Simulator of Simplified Three Cell Coupling Faults for Random Access Memories	2024	IEEE	1. Introduced ML3C algorithm and a simulator TCFS 2. The March ML3C algorithm targets the detection of single-port, static, and unlinked three-cell coupling faults within bit-oriented random access memory (RAM)

Data-driven Fault Detection of Multiple Open-circuit Faults for MMC Systems Based on Long Short-term Memory Networks	2024	IEEE	1. This study describes a fault detection technique based on long short-term memory (LSTM) that can be used to identify multiple open-circuit switch faults in modular multilevel converter (MMC) systems that have full-bridge sub-modules (FB-SMs) 2. In order to thoroughly extract the fault characteristics of MMC under various faults and operating settings, a multi-layer LSTM network is designed, and a Softmax layer identifies the different sorts of faults.
Long Short-Term Memory-Based Feedforward Neural Network Algorithm for Photovoltaic Fault Detection Under Irradiance Conditions	2024	IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT	1. In this study, PV defects included open circuit faults, short circuit faults, line-to-line (L-L) faults, mismatch faults (MF), partial shading (PS) faults, and inverter faults (IF). On the other hand, pressing matters pertaining to PV systems may be lessened if these difficulties are promptly and efficiently addressed and resolved. This article suggests a feedforward neural network (FFNN) technique based on long short-term memory (LSTM) that solves this problem by utilizing linear regression (LR), decision trees (DTs), and support vector machines (SVMs).
Deep Learning Based Relay for Online Fault Detection, Classification, and Fault Location in a Grid-Connected Microgrid	2023	IEEE	1. This article presents a first attempt at the online fault detection, fault classification, and fault site identification of a grid-connected Micro-grid (MG) system. To address the problems with the current techniques, a Long Short Term Memory (LSTM) network based on deep learning algorithms is proposed 2. combination of an LSTM network and a feed-forward neural network (FFNN) with a back-propagation algorithm (BPA) is suggested to find the location

Down Shift: Tuning Shift Reduction With Reliability for Racetrack Memories	2023	IEEE TRANSACTIONS ON COMPUTERS	1. In this research, a collaborative design for location error corrections and generalized data placement in RTM is presented. In order to increase self-accesses and reduce shifts, we introduced Down Shift, which examines the lifespans of memory items and directs them to DBCs. We presented GROGU, a unique, more versatile, and more effective reliability technique for position error correction in RTMs, based on realistic TR distances.
LCHC-DFT: A Low-Cost High-Coverage Design-for-Testability Technique to Detect Hard-to-Detect Faults in STT-MRAMs in the Presence of Process Variations	2022	IEEE TRANSACTIONS ON DEVICE AND MATERIALS RELIABILITY	1. In order to enhance the detection of hard-to-detect (HtD) faults in STT-MRAMs, this study suggests a high-coverage, low-cost design-for-testability (DFT) scheme. 2. The concept involves creating a voltage imbalance within the sensing amplifier (SA). Due to this discrepancy and the cell's faults the SA may be biased, producing an inaccurate read output that reveals the HDD flaws.
A Novel Relaying Scheme Using Long Short Term Memory for Bipolar High Voltage Direct Current Transmission Lines	2021	IEEE	1. This study presents a unique relaying technique that detects, identifies the fault pole, and estimates the fault location for bipolar line commutated converter (LCC) high voltage direct current (HVDC) transmission lines. The rectifier end DC current and voltage data are used to extract features for the scheme. For completing various relaying tasks, a deep learning technique called long short term memory (LSTM) has been developed as a predictor and classifier. The fault location estimation module (FL), the fault pole identification module (FI), and the fault detection module (FD) (LSTM-FD) are the three modules that have been designed.

Effective Spare Line Allocation Built-in Redundancy Analysis With Base Common Spare for Yield Improvement of 3D Memory	2021	IEEE	1. In this paper, a fast and small-area built-in redundancy analysis (RA) for the post-bond repair process in 3D memory is proposed. 2. The proposed BIRA improves the efficiency of spare lines with two complementary spare resource structures, achieving a short repair time and high repair rate.
BIST-Based Fault Diagnosis for PCM With Enhanced Test Scheme and Fault-Free Region Finding Algorithm	2021	IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS	1. This paper suggested a BIST-based test scheme as a solution to the challenges of high cost and low efficiency in PCM testing. 2. The following are the primary contents: a BIST-based test scheme with FFR finding method; an improved march test algorithm for bit-oriented PCM; and a converted march test technique for word-oriented PCM. 3. The suggested March-BOPCM improves the identification of PD issues. Along with adding the ability to detect possible IPD faults for word-oriented PCM with various array architectures 4. The modified March-WOPCM maintains its original detection capabilities. Additionally, to identify a continuous FFR in tested memory for the storing of identified fault information

MLParest: Machine Learning based Parasitic Estimation for Custom Circuit Design	2020	IEEE	1. A novel machine learning based parasitic estimation(MLParest) method for pre-layoutcustom circuit design is presented.It reduces the error between pre- layout and post-layout circuit simulation from 37% to 8% on average for different measurements across a variety of analog circuits. 2. MLParest can thus greatly reduce the number of iterations between pre-layout and post-layout design phases. The key contributions of this work are a machine learning based approach to parasitic estimation and a push-buttonmodel training framework,scalable across different
Rand Shift: An Energy-Efficient Fault-Tolerant Method in Secure Nonvolatile Main Memory	2020	IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS	1. In this paper, the authors suggested a technique to handle hard errors in non-volatile memory cells that makes use of rotational shift operation and the randomization characteristic of AES encryption. This approach, known as RandShift, benefited from low energy usage and a straightforward hardware implementation. It reduced the requirement to use effective error correction techniques like ECC and ECP.
Analog Integrated Circuits and Signal Processing	2020	Springer (SCI)	1. wavelet-based transientsupply current testing with modified March sequence exploiting read equivalent stress (RES) is introduced for fault detection in SRAM Cell. 2. Proposed a solution that ensures a minimum test timefor the detection of open defects in SRAMs 3. Proposed work solely relieson the hardware implementation for the faultdetection the proposed technique reduces design parameters, such as the area overhead, power consumption, hardware complexity and performanceoverhead

Chapter 3

Proposed Parasitic Extraction Method

As technology develops toward downsizing, dense eSRAMs with high error rates may result. Memory and SoC yield suffer as a result. As a result, some form of solution is necessary, one that is free of technology differences and independent of the fault model adopted. If the test methods are based on a fault model, the fault coverage as well as the test duration are limited. Another disadvantage of modern testing procedures is that they do not account for the effects of parasitic memory effect, resulting in an incomplete test. To that end, researcher suggested a testing method for embedded SRAMs that uses parasitic R, C extraction from a fault-induced architecture to detect extreme faults.

Although significant amount of research efforts has been put into the area of developing an effective and economic fault model for SRAM testing, the testing methodologies for highly integrated SRAM designs have not been fully discussed, and no studies/research carried out in this direction as per the reviewed literature. In this scenario, the researcher presents an investigational analysis on the effects of short defects injected in SRAM core cell.

Taking the effect of spot defect on layout, and by injecting them into SRAM core cell, the influence of spot defects can be studied [116]. Either the defect injection in the circuit is made by an open wire, a short between wires, or missing contacts etc. In this work, only node-to-node short defects are considered. By looking at the defect analyses available in the literature.

The foregoing work is done by considering single defect injected into SRAM core cell as there is less probability of occurrence of multiple defects in such a small cell. However, for experimenting purpose, the defects with multiple node open faults also considered in this work.

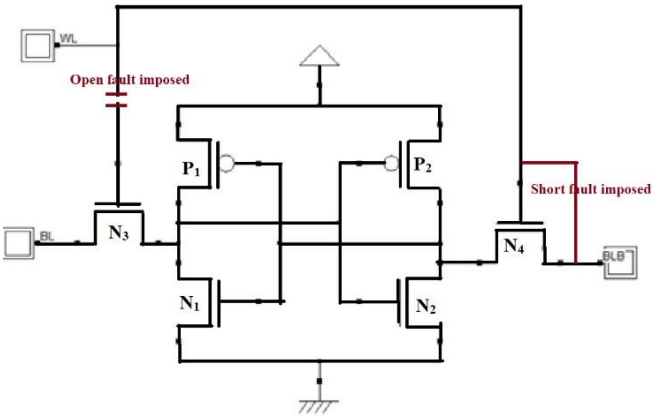
3.1 Parasitic Extraction Method Test Approach Using Microwind

Dealing with process variation in SRAM cells is a major design challenge. Failures induced by the manufacturing process in SRAM cells commonly result in destructive reads, unsuccessful writes, and an increase in read and write access time, as well as hold failures. The amount of failures is determined by SRAM operating circumstances such as power supply and frequency. There has been a lot of past work done on memory testing, and the most of it has been done using various March Algorithms. However, no method is suggesting the correct fault model, which stands against to process variation problems. As and when technology advancements are taking place, the existing test methods need to be re-modelled at the cost of test time development. The process of re-modelling of existing March algorithms for better fault coverage further enhances the test time complexity.

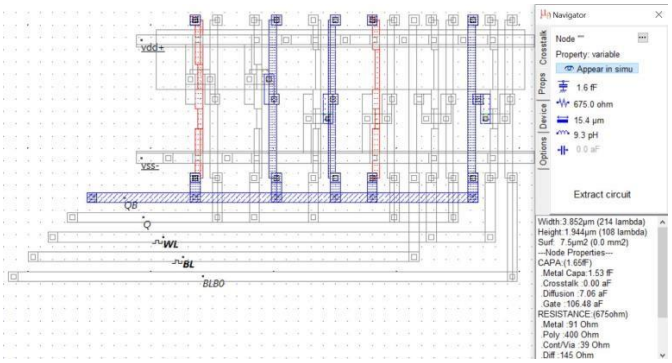
Hence, the researcher proposed an operational method, which characterizes the failures of SRAM using range of parasitic resistance and capacitance values resulting in the research titled as parasitic extraction, carried out in the style of Inductive Fault Analysis Method (IFA), but in reverse manner. The researcher has considered in this study the word 'inductive', precisely meaning the higher-level fault information is induced from lower level defects [117,118]. In Inductive Fault Analysis process, defects introduced in the layout are mapped to circuit level faults, whereas in the parasitic extraction method, circuit level faults are mapped to layout level defects

The steps involved in the proposed parasitic extraction method are

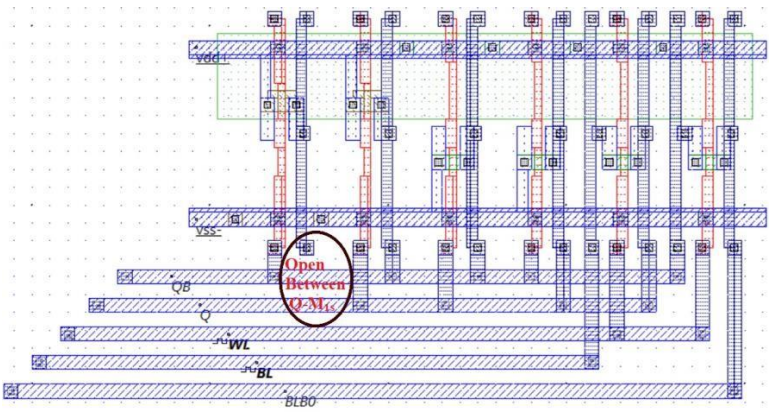
i) Model the circuit with fault imposed



ii) Using Layout Extract Parasitic R, C values for fault free SRAM.

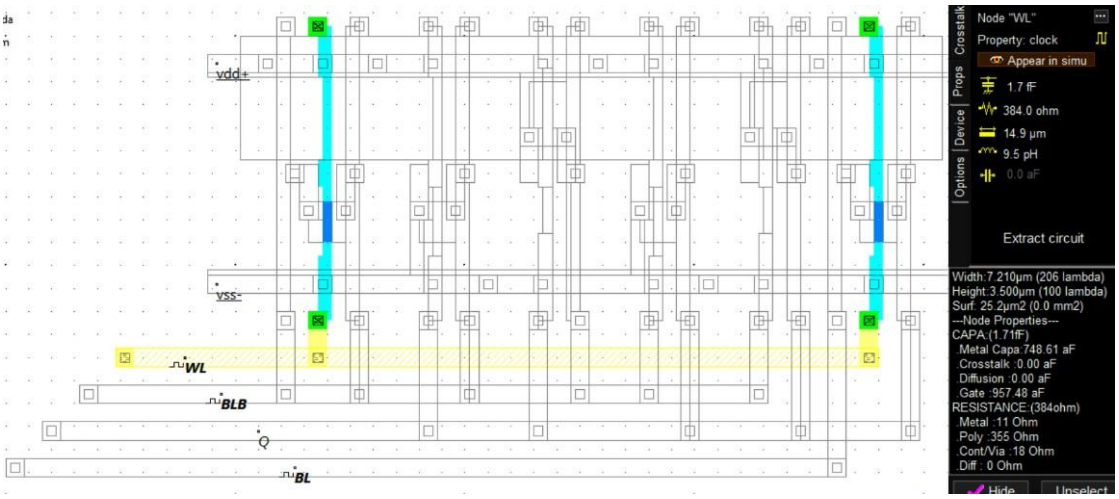


iii) Extract the defect induced layout from the fault model circuit



iv) Observe the defects in terms of short/open or missing of wires.

v) Extract and collect parasitic R and C from each faulty layout and compare them with proto typed fault free layout. Any deviation between R, C values of faulty and fault free cell at each node gives the fault information



vi) Use collected parasitic R,C values for exhaustive testing in real world test environment

When integrated circuits are manufactured, there is naturally occurring variance in the properties of transistors (length, breadth, and oxide thickness). As device sizes are often affected by the lithography process, the difference becomes a bigger percentage with lower technologies. Such variations will affect the stability of 6T SRAM cells and may lead to formation of hidden/undefined faults.

The main intension of this work is to develop a test method for testing Embedded SRAMs at core level for maximum fault detection including undetected faults. In this process three technology levels are considered for designing fault models, such as 45nm, 32nm and 7nm the corresponding process parameters are shown in table.

Table-3.1: Technological Evaluation (Source: microwind.net)

Technology Node	130nm	90nm	65nm	45nm	32nm	22nm	16nm	11nm
First Production	2001	2003	2005	2007	2009	2011	2013	2015
Effective Gate Length	70nm	50nm	35nm	30nm	25nm	18nm	12nm	9nm
Gate Material	Poly	Poly	Poly	Metal	Metal	Metal	Dual?	Triple?
Gate Dielectric	SiO_2	SiO_2	SiON	High K	High K	High K	High K	High K
Raw Mgate/mm^2	0.25	0.4	0.8	1.5	2.8	5.2	9.0	16.0

Table 3.2. Comparison of Transistor Parameter for different technologies

Parameter	180nm	120nm	90nm	65nm	45nm	32nm	14nm	7nm
VDD(V)	2	1.2	1	1	1	1	0.8	0.8
Tdelay(ns)	0.03	0.03	0.005	0.005	0.003	0.0025	0.0016	0.0012
THvDelay(ns)	0.1	0.06	0.02	0.01	0.008	0.007	0.007	0.004
THsDelay(ns)	0.6	0.02	0.004	0.003	0.002	0.002	0.005	0.002
TWireDelay(ns)	0.1	0.07	0.005	0.002	0.0015	0.0014	0.001	0.001
Tcurrent(mA)	0.6	0.5	0.1	0.1	0.08	0.07	0.03	0.04
ML(um)	0.18	0.12	0.1	0.07	0.05	0.03	0.016	0.007
MHvL(um)	1.5	0.36	0.3	0.2	0.18	0.036	0.01	0.01
MNW(um)	1.5	1	0.5	0.3	0.3	0.08	0.048	0.024
MPW(um)	1.5	2	1	0.5	0.5	0.108	0.048	0.024

The numbers 45nm, 32nm, 7nm are representing the minimal channel length that can be fabricated. The minimal length of the MOS transistor channel between the drain and source is defined as the feature size of any semiconductor technology. For the design of embedded SRAM, three technologies have been selected 45nm, 32nm, and 7nm technologies. Table 3.2, gives the comparison and the overview of the key parameters like supply voltage, delays, current, and length and width of the transistors for the different technology nodes. In the analysis of the parameters, the researcher have considered three modes of operation, Standard, High Voltage, and High Speed. For example in the calculation of delays. Tdelay represents standard time delay, the delay represents the delay in high voltage mode and THs represent high-speed mode.

The numbers 180nm, 120nm, 90nm, 65nm, 45nm, 32nm, 14nm, and 7nm are representing the minimal channel length that can be fabricated.

3.2. Proposed Fault Model with Open Defects in Single Cell SRAM

Fig.3.1 shows 6T SRAM cell with five main nodes Q, QB, BL, BLB, and WL. Out of which, Q, QB are internal nodes through them the cell state can be monitored and WL, BL and BLB are external nodes through these writing and reading operations can be performed. Two more nodes are supply and ground nodes V_{DD}, Vss respectively. All possible shorts between internal and external nodes are considered in the proposed fault model.

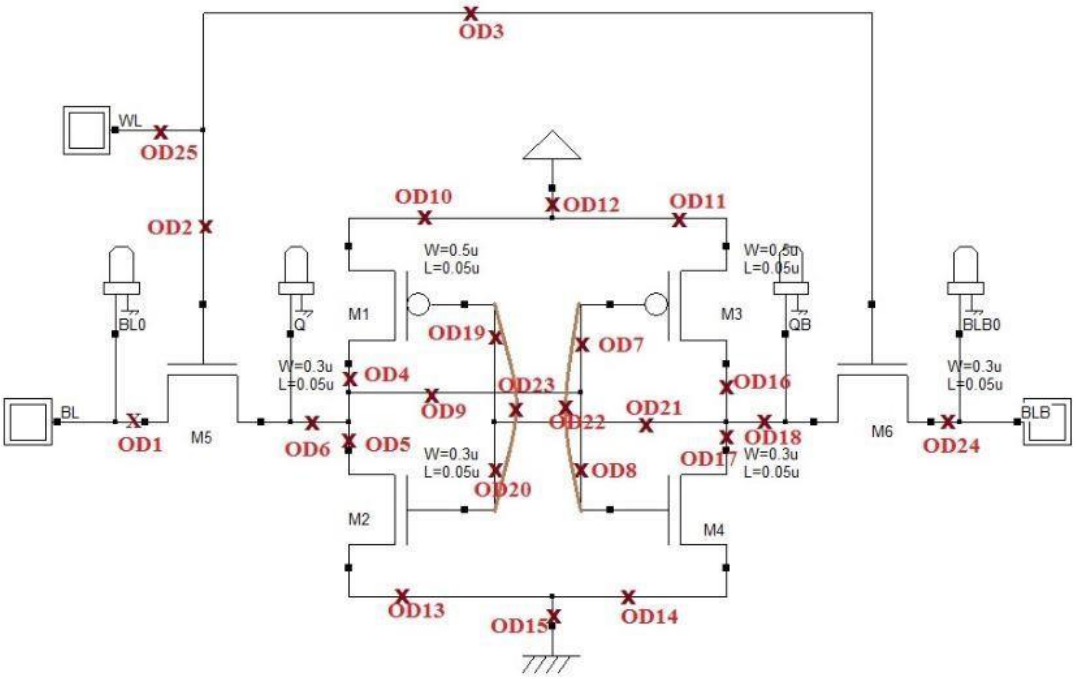


Fig 3.1. Proposed Fault Model for open defects

Fig 3.1. Shows the proposed complete fault model for open defects. In the figure OD depicts the Open Defect. Here the researcher consider the node to node open defects, and few multi node open defects. Fault is nothing but deviation of the output from the correct output. As the technology advances, the VLSI chip contains millions of the transistors on the single IC. The number of physical defects is too many. It is very challenging almost impossible to analyze and locate all the faults. Therefore, there is a need to develop a method to test memory, which is independent of the technology

variation. DC parametric testing, AC parametric and functional testing were proposed prior to 1990. However, all these algorithms are mathematical in nature. After that number of fault models are developed, which will be used to construct algorithms and asses their quality, generally march algorithms are used to detect the faults in embedded memory. The number of nodes increases in March algorithms as test complexity increases. Many existing fault models does not consider the parasitic effects, which causes the many undetectable faults. Due to technology advancement, scale down technologies influence the parasitic effect, which will cause to the additional faults. The researcher have proposed a new parasitic extraction method for fault detection along with identification of fault location. This is a layout dependent method which gives 100 percent fault coverage, irrespective of the technology variation.

Table 3.3. Shows the test results of 32nm and 7nm technologies. The parasitic variations are compared with that of fault free SRAM. The proposed parasitic extraction method identifies the type of fault along with its location independent of the technology selected.

As shown in the fig. 3.1. This experiment carried totally 25 open defect fault models. For each fault model researcher have extracted node resistance and capacitance values, and compared with resistance and capacitance values of fault free SRAM. Table 3.7 shows the extracted parasitic R and C values for both fault free and fault models at nodes Q, QB, WL, BL, BLB, VDD, VSS.

3.3 Faults Identified for Open Defects for different technologies

The overall single cell SRAM fault model using node-to-node open defects, resulted from three technologies are 63. Microwind (3.9version) tool was used to implement fault models. The list of the faults and their acronyms are shown in table 3.3

Open resistive faults are produced during the manufacturing time of 6T SRAM. The behaviour of the memory cell may modify by these open faults. In the proposed parasitic extraction method, researcher have extracted the Resistance(R) and Capacitance(C) values for fault free SRAM cell, then researcher have imposed open defects between different nodes, then researcher have extracted the R and C values for faulty SRAM cell. By comparing these R and C values researcher can identify the any

fault. In the proposed method researcher have identified faults like No Access Fault (NAF), Undefined Read Fault (URF), Undefined Write Faults (UWF), and Transition Faults (TF).

Table 3.3: Open Fault Dictionary for Single Cell 6T SRAM

S. No	***Defect Model***	***Open Defect Location***	***Fault Type***	
			32nm	***7nm***
1	OD_1	BL-M_{5S}	NAF	NAF
2	OD_2	WL- M_{5G}	NAF	NAF
3	OD_3	WL- M_{6G}	URF	URF
4	OD_4	Q-M_{1D}	UWF1	UWF1
5	OD_5	Q-M_{2D}	UWF0	UWF0
6	OD_6	Q-$M_{1D}M_{2D}$	NAF	NAF
7	OD_7	Q-M_{3G}	TF	UWF0, URF0
8	OD_8	Q- M_{4G}	TF	UWF1, URF1
9	OD_9	Q-$M_{3G}M_{4G}$	NAF	NAF
10	OD_{10}	V_{DD}-M_{1S}	UWF1	UWF1
11	OD_{11}	V_{DD}-M_{3S}	TF	UWF0, URF0
12	OD_{12}	V_{DD}-$M_{1S}M_{3S}$	UWF,URF0	UWF,URF0
13	OD_{13}	V_{SS}-M_{2S}	UWF0	UWF0
14	OD_{14}	V_{SS}-M_{4S}	TF	UWF1, URF1
15	OD_{15}	V_{SS}-$M_{2S}M_{4S}$	UWF, URF1	UWF, URF1
16	OD_{16}	Q_B - M_{3D}	TF	UWF0, URF0
17	OD_{17}	Q_B - M_{4D}	UWF1,URF1	UWF1,URF1
18	OD_{18}	Q_B_$M_{3D}M_{4D}$	URF0, UWF	URF, UWF0
19	OD_{19}	Q_B_M_{1G}	UWF1	UWF1
20	OD_{20}	Q_B_M_{2G}	UWF0	UWF0
21	OD_{21}	Q_B_$M_{1G}M_{2G}$	UWF	UWF
22	OD_{22}	M_{1G}_M_{2G}	UWF	UWF
23	OD_{23}	M_{3G}_M_{4G}	NAF	NAF
24	OD_{24}	BLB - M_{6D}	URF	URF
25	OD_{25}	WL-$M_{5G}M_{6G}$	NAF	NAF

3.3.1 No Access Fault:

The memory cell said to you have No Access Fault, if the cell is not accessible. In this case, researcher cannot perform any write operation, any read operation of the cell. In the open fault analysis, No Access Fault occur for the fault defects OD_1(BL-M_5S), OD_2(WL- M_5G), OD_6(Q-M_1DM_2D), OD_9(Q-M_3GM_4G) and OD_{25}(WL-M_5GM_6G)

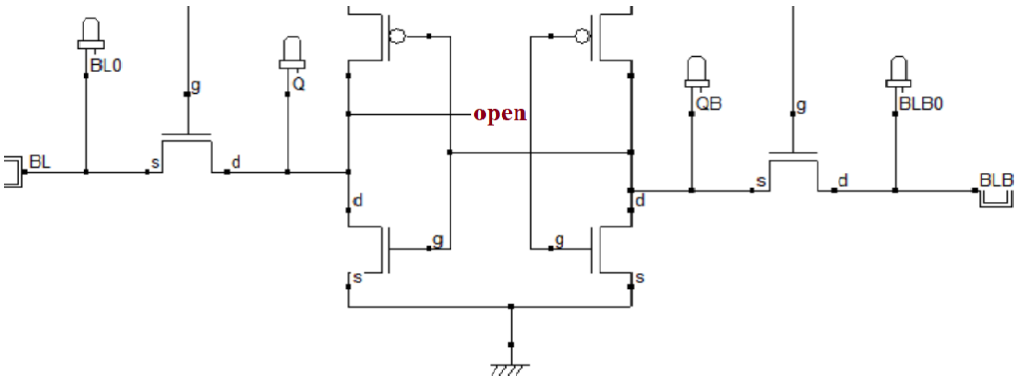


Fig 3.2(a). Fault Model for open defect between Q and M_3GM_4G

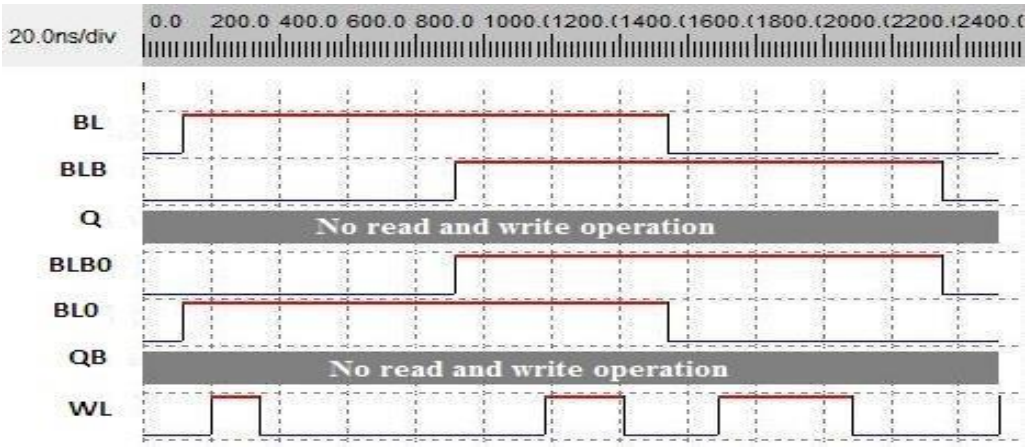


Fig.3.2 (b). Simulation results for No Access Faults

3.3.2. Undefined Read Fault:

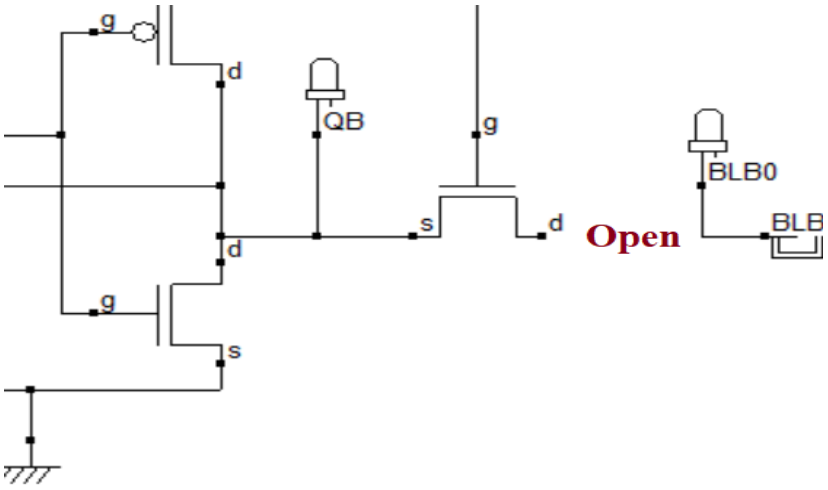


Fig 3.3(a). Fault Model for open defect between M_6D and BLB

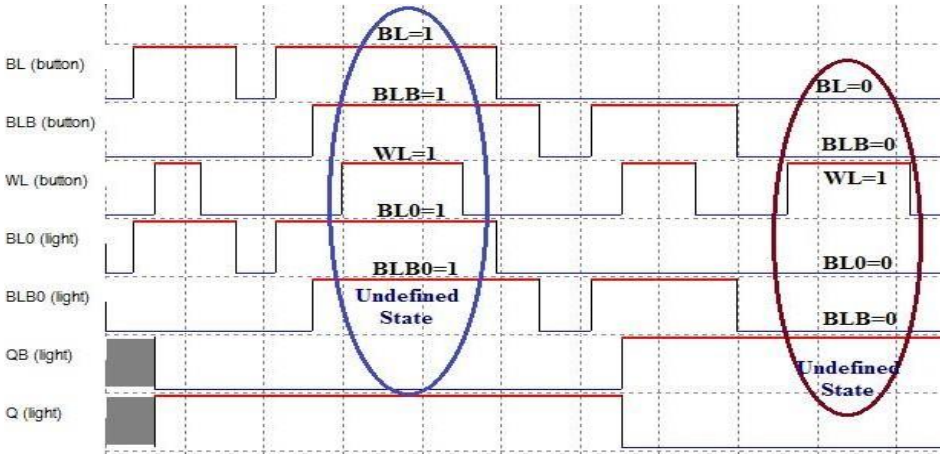


Fig.3.3(b). Simulation results for Undefined Read Fault at OD_3

The memory cell said to have Undefined Read Fault, if the cell is brought into an undefined state through a read operation. Undefined means, the cell state goes to neither '1' nor '0' with a read operation. This fault occurs for the open defects induced at OD_3(WL- M_6G), OD_7(Q-M_3G), OD_8(Q- M_4G), OD_{11}(V_{DD}-M_3S), OD_{12}(V_{DD}-M_1SM_3S), OD_{14}(V_{SS}-M_4S), OD_{15}(V_{SS}-M_2SM_4S), OD_{16}(Q_B-M_3D), OD_{17}(Q_B-M_4D), OD_{18}(Q_B_M_3DM_4D), OD_{24}(BLB - M_6S)

3.3.3 Undefined Write Fault:

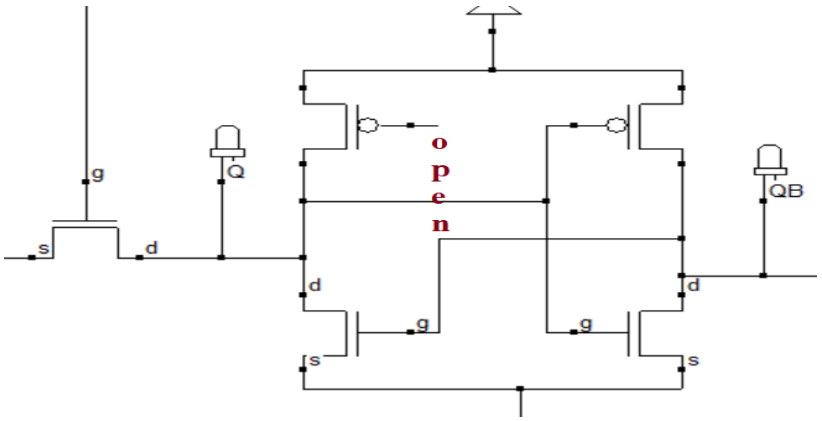


Fig 3.4(a). Fault Model for open defect between M_1G and M_2G

The memory cell said to have Undefined Write Fault, if the cell is brought to in an undefined state through a write operation. Undefined means, the cell state goes to neither '1' nor '0' with a read operation. This fault occurs for the open defects induced at OD_4(Q-M_1D), OD_5(Q-M_2D), OD_7(Q-M_3G), OD_8(Q- M_4G), OD_{10}(VDD-M_1S), OD_{11}(VDD-M_3S), OD_{12}(VDD- M_1SM_3S), OD_{13}(VSS-M_2S), OD_{14}(VSS-M_4S), OD_{15}(VSS-M_2SM_4S), OD_{16}(QB-M_3D), OD_{17}(QB-M_4D), OD_{20}(QB_M_2G), OD_{21}(QB_M_1GM_2G), D_{22}(M_1G_M_2G), OD_{19}(QB_M_1G), OD_{18}(QB_M_3DM_4D)

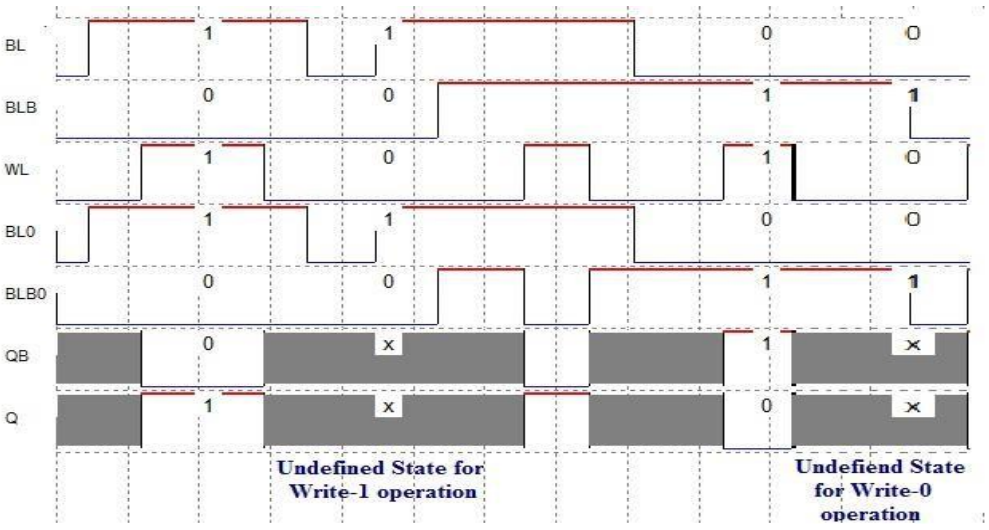


Fig.3.4 (b). Simulation results for Undefined Write Fault at OD_{21}

3.3.4 Transition Fault:

The memory cell is said to have a transition fault, if it fails to undergo a transition in a write operation. This functional fault model depends both on the initial stored value and the type of the operation (Read/Write) performed. researcher should be able to write '0' in the cell stored with '1' and vice versa. However, if the cell fails to perform a transition from its initial stored value, it manifests as TF.

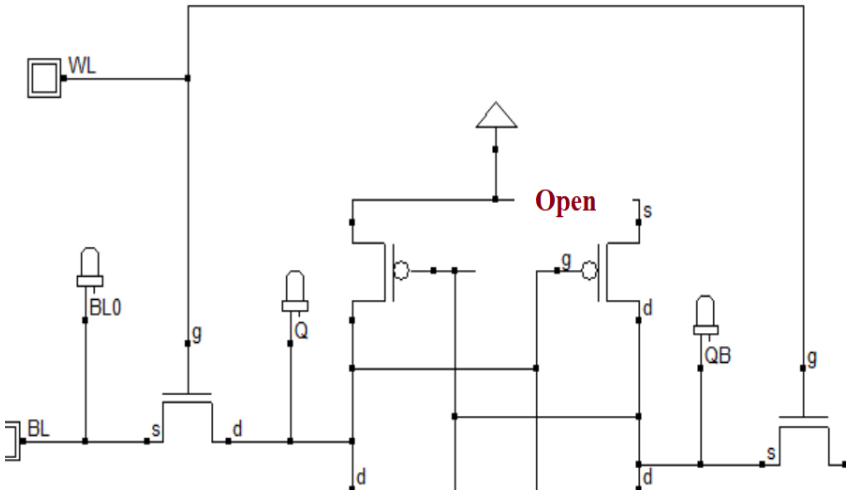


Fig 3.5(a). Fault Model for open defect between VDD and M_3S

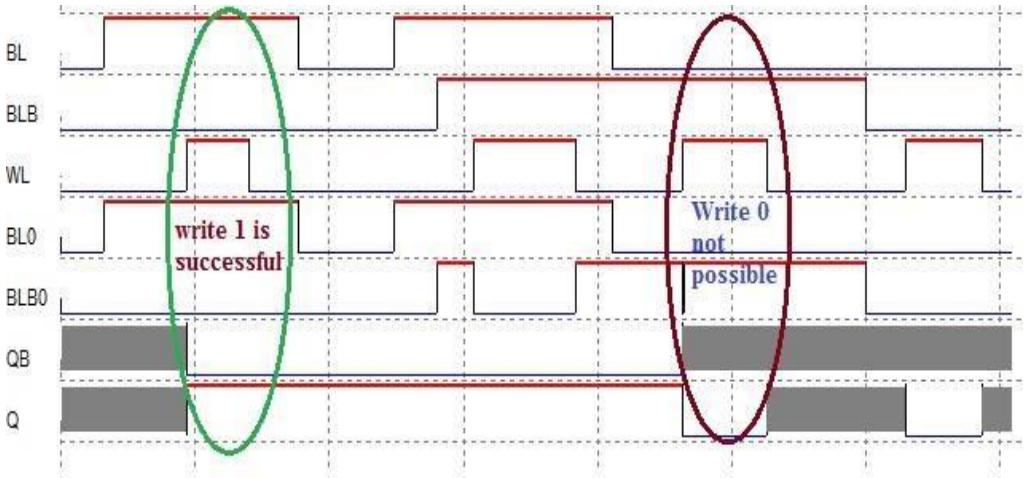


Fig.3.5 (b). Simulation results for Transition Fault

3.4 Extraction of Parasitic R, C at 6T-SRAM node points

The 6T SRAM layout is shown in Fig.3.6a. The parasitic R, C values are observed at each individual node (Q, QB, BL, BLB, and WL) using MW simulation environment and the corresponding process and the steps explained earlier sections. On selecting a particular node, it gives the total parasitic R, C at that node as shown in Fig.3.6b. Parasitic capacitance measured in fempto Farads (fempto $=10^{-15}$), is a combination of metal capacitance, crosstalk capacitance, diffusion capacitance, and gate capacitance. Similarly, parasitic resistance measured in ohms, comprises with metal resistance, poly resistance via resistance and diffusion resistance. The other two values indicates the node connectivity i.e., length in terms of um (Micron Meter) and inductance L in terms of nH (nano Henry). Using 45nm technology, at node QB, the observed parasitic capacitance is 1.6 fF, parasitic resistance is 675 ohms, and node length is 15.4 um. The procedure is same for other two technologies 32nm and 7nm. Only parasitic R & C are considered throughout the fault detection process in this thesis.

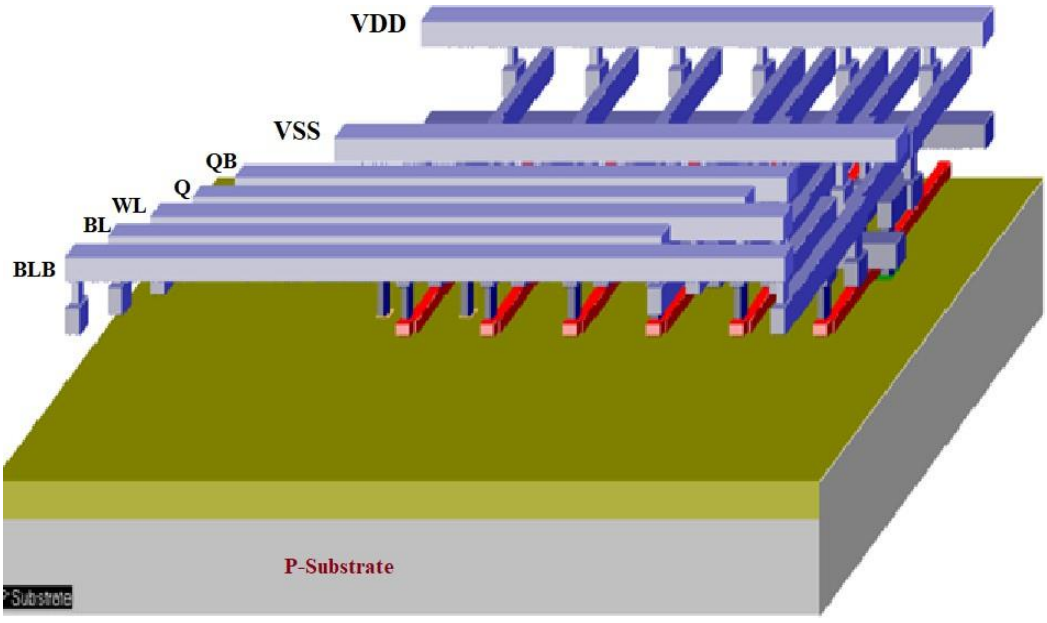


Fig 3.6a .3D View of Layout diagram for fault free 6T SRAM

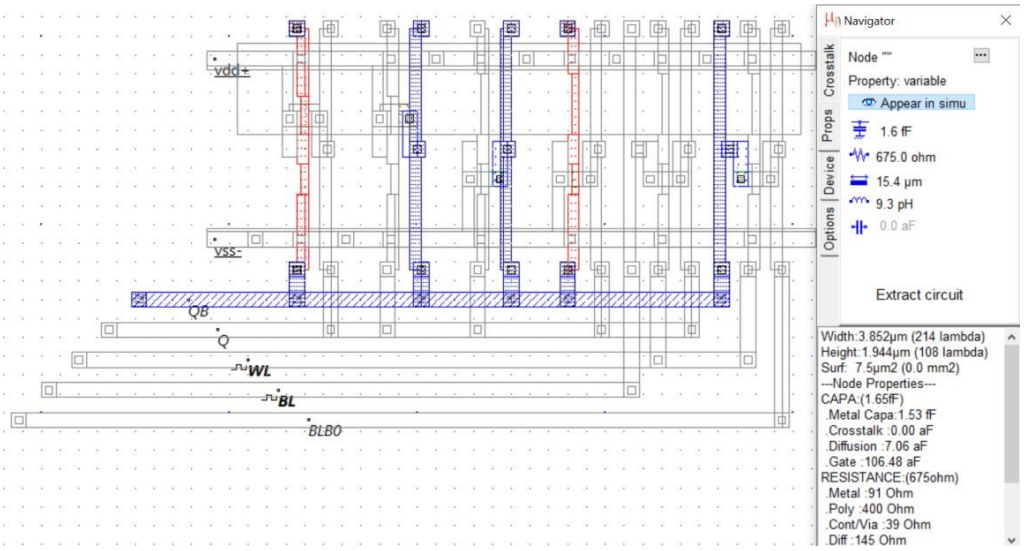


Fig 3.6b. Parasitic Extraction at node QB for fault free SRAM Cell

The parasitic R, C values at each node are observed and are shown in Table 3.4. Fault detection through parasitic R, C is accomplished, by comparing each fault model parasitics with fault free model.

Table 3.4. Parasitic R, C values of Fault Free SRAM Cell for different technologies

Input-output nodes	**Fault free SRAM Cell Parasitic R, C values**					
	45nm		**32nm**		**7nm**	
	R(Ω)	C(aF)	R(Ω)	C(aF)	R(Ω)	C(aF)
Q	6881	1900	677	1800	433	2900
QB	7585	1800	497	1500	1170	3100
WL	4712	663	421	791	180	1800
BL	1216	664	75	701	158	1100
BLB	240	354	79	637	54	783
VDD	6600	1900	31	313	2071	2700
VSS	2823	1300	13	313	402	1700

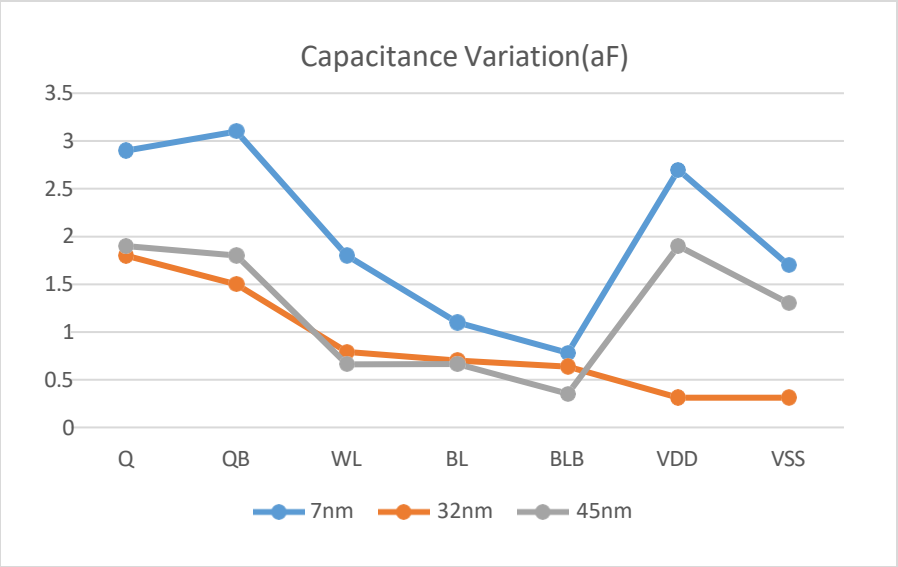


Fig 3.7a. Parasitic capacitance variation for different technologies for fault free SRAM Cell

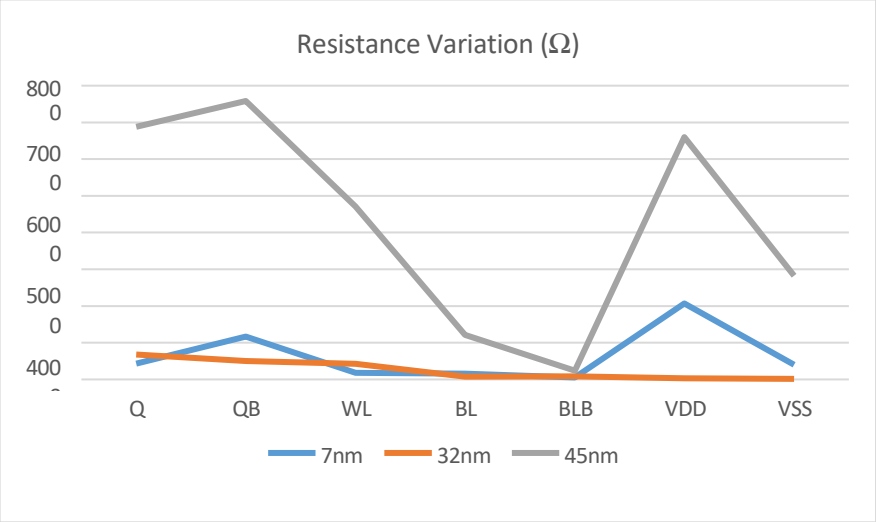


Fig 3.7b. Parasitic Resistance values for different technologies for fault free SRAM Cell

3.5 Fault Detection Using Parasitic R, C Extraction Method

3.5.1 Open Fault Detection for 32nm Technology

Four fault models are considered as shown in Table 3.5. Each fault model with corresponding fault affecting node is represented. For each fault model, layout is extracted, and the fault affecting nodes are identified on the layouts, as shown in Figures .3.8a, 3.8b, 3.8c, and 3.8d.

Table 3.5: Fault Models for all open faults

S.No	Open between nodes	Fault Occurs
1	BL-M_{5S}, WL- M_{5G}, Q-$M_{1D}M_{2D}$, Q-$M_{3G}M_{4G}$, M_{3G}_M_{4G}, WL-$M_{5G}M_{6G}$	NAF
2	WL- M_{6G}, V_{DD}-$M_{1S}M_{3S}$, V_{SS}-$M_{2S}M_{4S}$, Q_B - M_{4D}, Q_B_$M_{3D}M_{4D}$, BLB - M_{6D}	URF
3	Q-M_{1D}, Q-M_{2D}, V_{DD}-M_{1S}, V_{DD}-$M_{1S}M_{3S}$, V_{SS}-M_{2S}, V_{SS}-$M_{2S}M_{4S}$, Q_B - M_{4D}, Q_B_$M_{3D}M_{4D}$, Q_B_M_{1G}, Q_B_M_{2G}, Q_B_$M_{1G}M_{2G}$, M_{1G}_M_{2G}	UWF
4	Q-M_{3G}, Q- M_{4G}, V_{DD}-M_{3S}, V_{SS}-M_{4S}, Q_B - M_{3D},	TF

Node WL is effected due to defect model WL-M_{6G} (fault model for URF) shown in Fig.4.11a. Node BL is effected due to defect model BL-M_{5S} (fault model for NAF) as shown in Fig.4.11b. Similarly, node Q is effected with fault model Q-M_{1D}(fault model for UWF, URF) in Fig.3.8c. Two nodes QB are affected using defectQ_B-M_{3D} (fault model for TF) and the same is shown in 3.8d

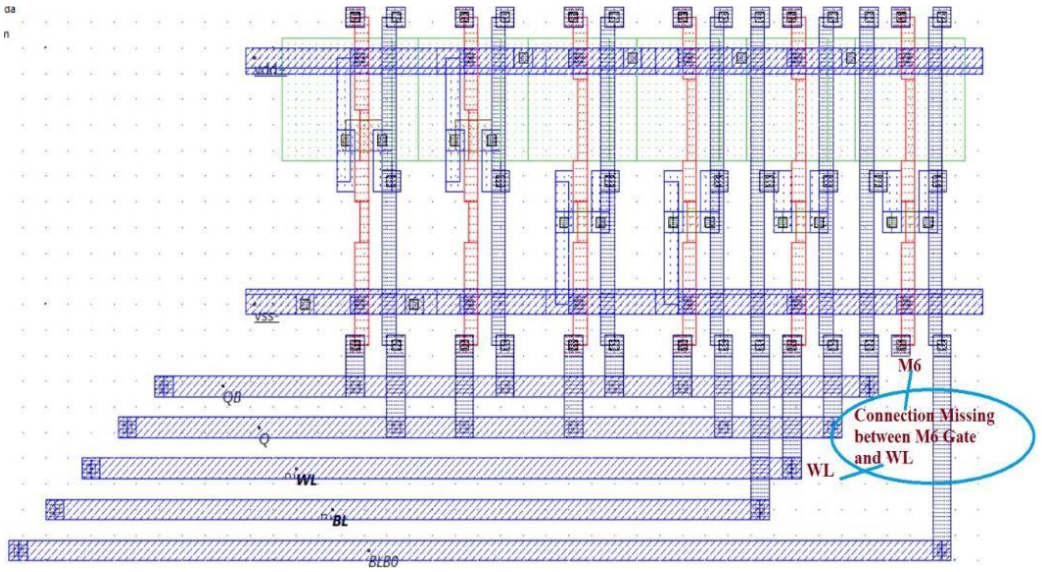


Fig 3.8a. Layout diagram for URF fault with open defect at WL-M_{6G}

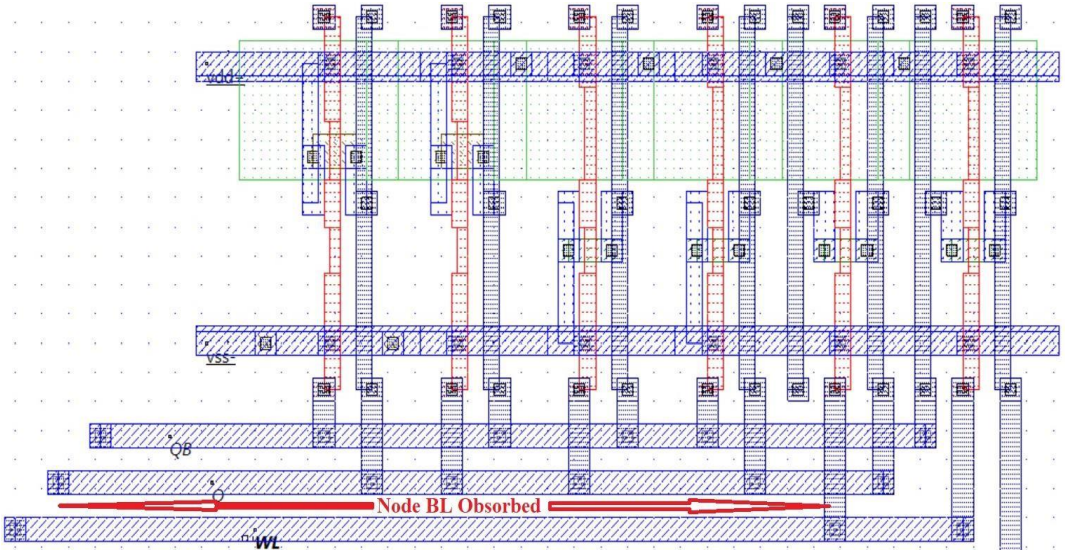


Fig 3.8b. Layout diagram for NAF fault with open defect at BL-M_{5S}

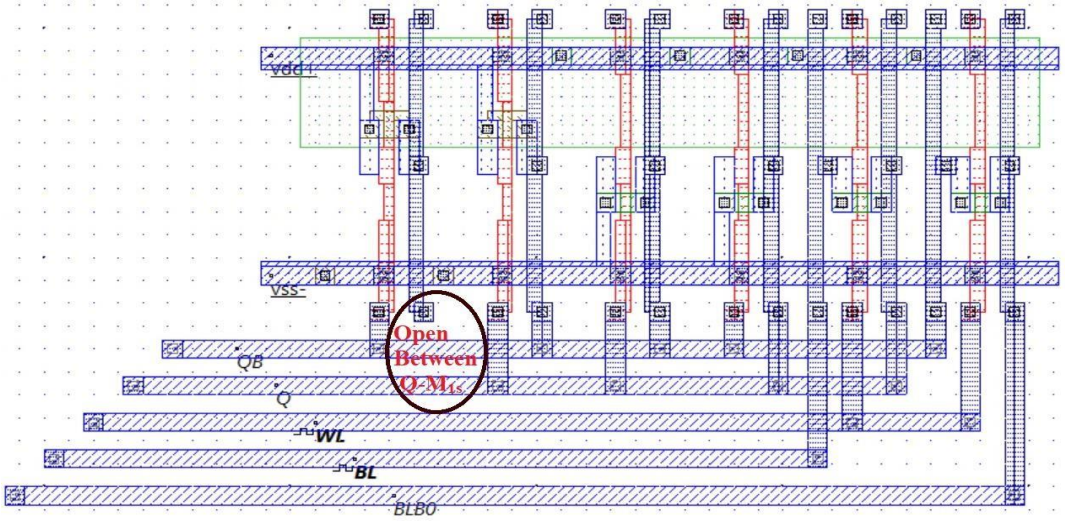


Fig 3.8c. Layout diagram for URF& UWF fault with open defect at Q-M_{1D}

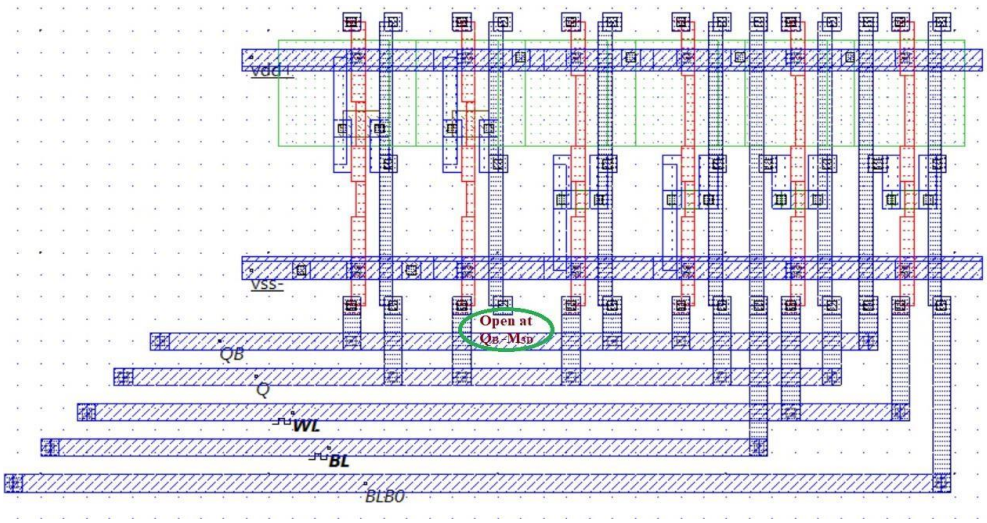


Fig 3.8d. Layout diagram for TF fault with open defect at Q_B-M_{3D}

Table 3.6 Parasitic R, C at affected nodes for chosen open fault models using 32nm Technology

S. No	Open Defect	Node Q		Node QB		Node WL		Node BL		Node BLB		Node VDD		Node VSS	
		Fault FreeC=1.7fF, R=800Ω		Fault Free C = 1.5fF,R=498 Ω		FaultFreeC = 0.77fF, R=296Ω		Fault FreeC = 0.62fF,R=71 Ω		Fault FreeC = 0.81fF,R=91 Ω		Fault FreeC = 0.31fF,R=13 Ω		Fault FreeC = 0.31fF,R=13 Ω	
		C in fF	R in Ω	C in fF	R in Ω	C in fF	R inΩ	C in fF	R inΩ	C in fF	R inΩ	C in fF	R inΩ	C in fF	R inΩ
1	BL-M5S(NAF)	1.80	805	1.50	498	0.78	296	NA	NA	0.82	91	0.31	13	0.31	13
2	WL-M6G(URF)	1.80	813	1.50	498	0.52	155	0.63	71	0.82	91	0.31	13	0.31	13
3	QB-M3D(TF)	1.80	813	1.20	392	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
4	QB_M2G(UWF)	1.80	813	1.40	362	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13

Figure 3.9b depicts a graphical representation of fault detection based on the variation in parasitic capacitance for 32nm technology. The open between BL and source of M_5 transistor causes to 'No Access Fault'. As shown in the fig.3.9b because of open defect the node BL is absorbed and the parasitic capacitance same at other nodes as Fault Free (FF) SRAM, except node BL.

Detection of faults using parasitic extraction method is achieved by comparing parasitic R and C values of fault model, with parasitic R and C values of fault free model. For example, if there is an open between node QB and drain of transistor M_3, which causes 'Transistor Fault', this we can detect by extracting R and C values at node QB and drain of the transistor M3. The parasitic C and R values at QB for fault free SRAM is 1.8fF and 813ohms respectively, when fault induced between node QB and drain of transistor M3, the parasitic C and R values changes to 1.2fF and 392ohms respectively. Whereas at other nodes, no change in the parasitic R and C values. Similarly considering the 'Undefined Write Fault' corresponding to open defect modeled by $QB\text{-}M_{2G}$. For this fault the capacitance value changes from 1.5fF to 1.4fF and the resistance value changed to 498Ω to 362Ω at other nodes no change in the corresponding parasitic R,C values. The same explanation is applicable for 'Undefined Read Fault' corresponding to open defect modeled by open between $WL\text{-}M_{6G}$, in which the parasitic capacitance variation affect more at node WL, whereas at remaining nodes they remain same as that of Fault Free.

Fault detection with resistance variation at various node points is shown in Fig.3.9a. The graph is drawn by taking all the faults on X-axis and parasitic resistances on Y-axis, and the resistance measured in ohms. The same explanation of detection of faults using variation in the 'parasitic capacitance' is valid, and holds for fault detection using 'parasitic resistance' variation. As a result, at least one node with a fault will be identified with its accompanied parasitic values, and this may be easily discovered during the testing using parasitic values.

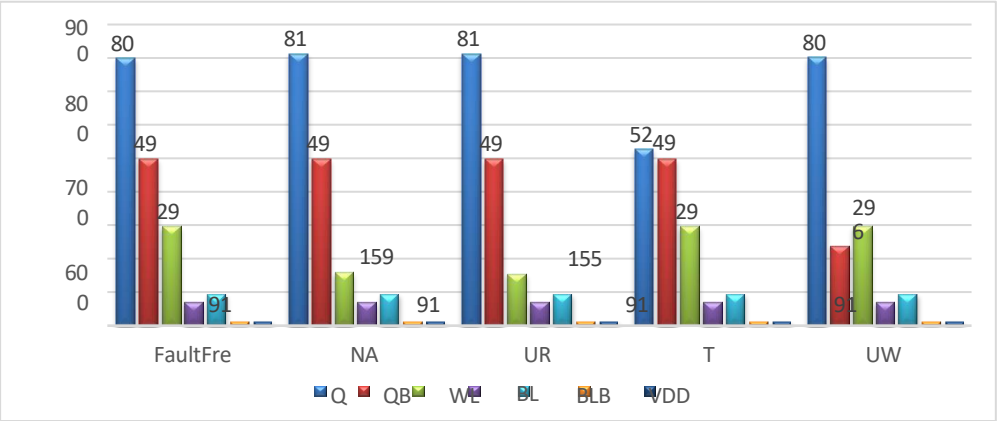


Fig 3.9a. Variation in Resistance for different faults

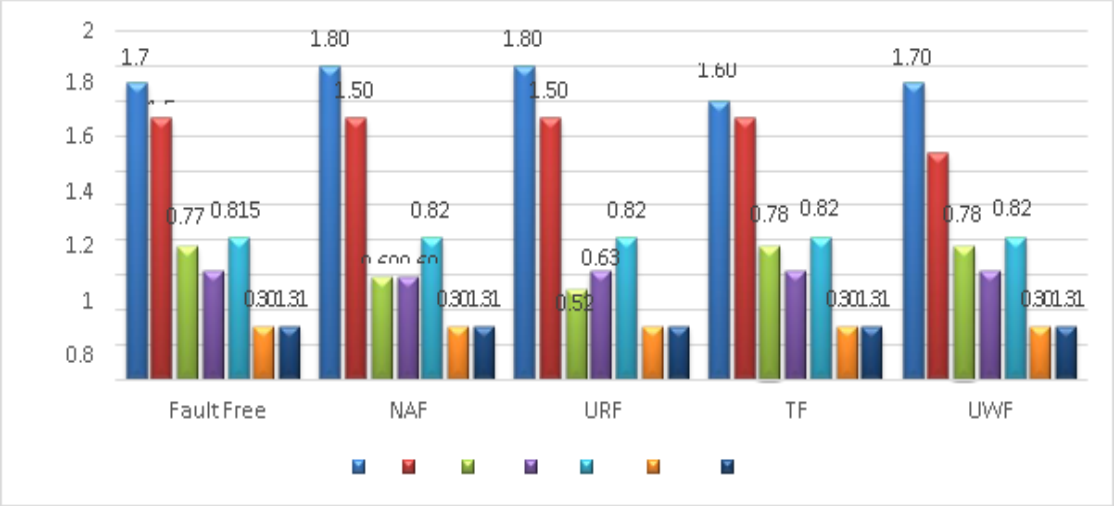


Fig 3.9b. Variation in Capacitance for different faults

Table 3.7. Extracted R, C Values Single 6T-SRAM cell for all open defects using 32nm Technology

S.No	Open Defect	Node Q		Node QB		Node WL		Node BL		Node BLB		Node VDD		Node VSS	
		Fault Free C = 1.7fF, R=800Ω		Fault Free C = 1.5fF, R=498 Ω		Fault Free C = 0.776fF, R= 296Ω		Fault Free C = 0.626fF, R=71Ω		Fault Free C = 0.815fF, R= 91Ω		Fault Free C = 0.313fF, R=13Ω		Fault Free C = 0.313fF, R= 13Ω	
		C (fF)	R (Ω)	C (fF)	R (Ω)	C (fF)	R (Ω)	C (fF)	R (Ω)	C (fF)	R (Ω)	C (fF)	R (Ω)	C (fF)	R (Ω)
1	BL-M5S	1.80	805	1.50	498	0.78	296	NA	NA	0.77	87	0.31	13	0.31	13
2	WL- M5G	1.80	813	1.50	498	0.60	159	0.60	70	0.82	91	0.31	13	0.31	13
3	WL- M6G	1.80	813	1.50	498	0.52	155	0.63	71	0.82	91	0.31	13	0.31	13
4	Q-M1D	1.40	682	1.50	498	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
5	Q-M2D	1.50	755	1.50	498	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
6	Q-M1DM2D	0.68	80	1.50	498	0.73	293	0.63	71	0.82	91	0.31	13	0.31	13
7	Q-M3G	1.60	527	1.50	498	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
8	Q- M4G	1.60	551	1.50	498	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
9	Q-M3GM4G	1.30	257	1.50	498	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
10	VDD-M1S	1.70	711	1.60	511	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
11	VDD-M3S	1.80	813	1.60	511	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13

12	VDD-M1SM3S	1.70	711	1.60	511	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
13	VSS-M2S	1.70	800	1.60	511	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
14	VSS-M4S	1.70	800	1.60	511	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
15	VSS-M2SM4S	1.80	813	1.60	511	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
16	QB - M3D	1.80	813	1.20	392	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
17	QB - M4D	1.80	813	1.30	444	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
18	QB_M3DM4D	1.70	803	0.70	75	0.78	296	0.60	70	0.77	87	0.31	13	0.31	13
19	QB_M1G	1.60	793	1.50	375	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
20	QB_M2G	1.80	813	1.40	362	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
21	QB_M1GM2G	1.70	803	1.30	239	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
22	M1G_M2G	1.60	793	1.30	239	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
23	M3G_M4G	1.30	257	1.50	498	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
24	BLB - M6S	1.80	805	1.50	498	0.78	296	0.63	71	0.82	91	0.31	13	0.31	13
25	WL-M5GM6G	1.80	813	1.50	498	NA	NA	0.60	70	0.77	87	0.31	13	0.31	13

The complete fault model dictionary for proposed fault models with parasitic R and C values using 32nm technology for 6T SRAM cell is shown in table 3.7. The complete fault model dictionary gives all fault model parasitic values taken node Q, QB, WL, BL, BLB, VDD and VSS. The variations are further compared with fault free. At which node the fault is imposed that corresponding node parasitics are affected in particular with high parasitic R, C variation.

By considering input/output nodes alone, the parasitic values of few fault models exhibiting their unique parasitic R, C variation at input/output nodes with 32nm technology. The same explanation holds for the 45 nm technology also.

3.5.2 Open Fault Detection for 45nm Technology

Fault detection through parasitic C and R at 45nm for the defect models WL- M_5G, Q-M_3G, M_1G-M_2G and BLB-M_6S is analogous to fault detection at 32nm. Table 3.8 shows the parasitic values of chosen fault models in 45nm technology, and it is observed that the fault model is resulting different parasitic values from one technology to other technology

Table 3.8 Parasitic R, C at affected nodes for chosen open fault models for 45nm Technology

Node	Fault free		WL- M5G (NAF)		Q-M3G (TF)		M1G_M2G (UWF)		BLB - M6S (URF)	
	C(fF)	R(KΩ)	C(fF)	R(KΩ)	C(fF)	R(KΩ)	C(fF)	R(KΩ)	C(fF)	R(KΩ)
QB	**2.9**	**18.96**	2.90	18.96	2.90	18.96	2.30	13.72	2.80	18.90
Q	**3.3**	**20.18**	3.30	20.18	2.80	16.13	3.20	20.15	3.10	20.10
WL	**0.88**	**2.66**	0.26	1.36	0.88	2.66	0.88	2.66	0.88	2.66
BL	**1**	**2.6**	1.00	2.60	1.00	2.60	1.00	2.60	1.00	2.60
BLB	**0.61**	**3.29**	0.61	3.29	0.61	3.30	0.61	3.30	NA	NA
VDD	**2.4**	**12.03**	2.40	12.03	2.40	12.03	2.40	12.03	2.40	12.03
VSS	**1.9**	**6.11**	1.90	6.11	1.90	6.11	1.90	6.11	1.90	6.11

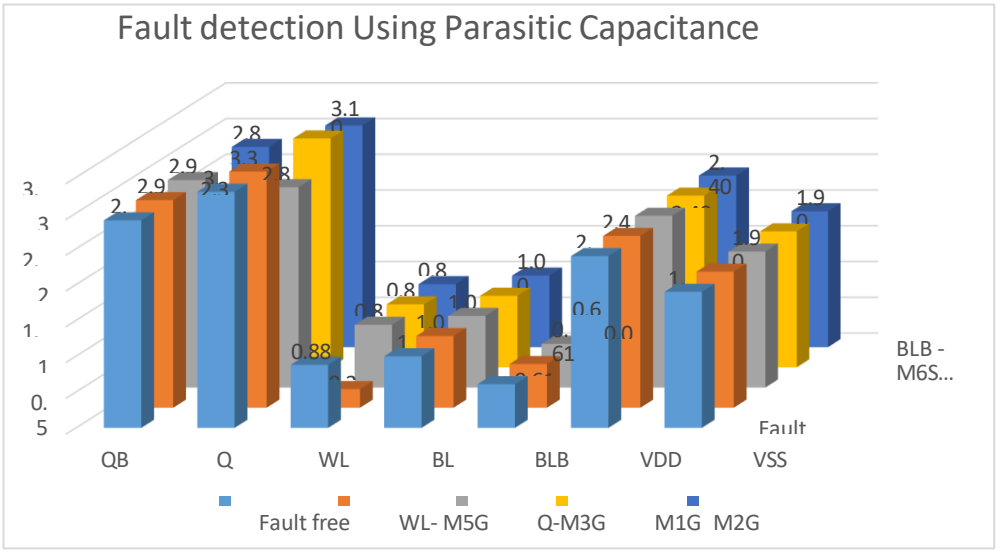


Fig 3.10a. Variation in Capacitance for different faults

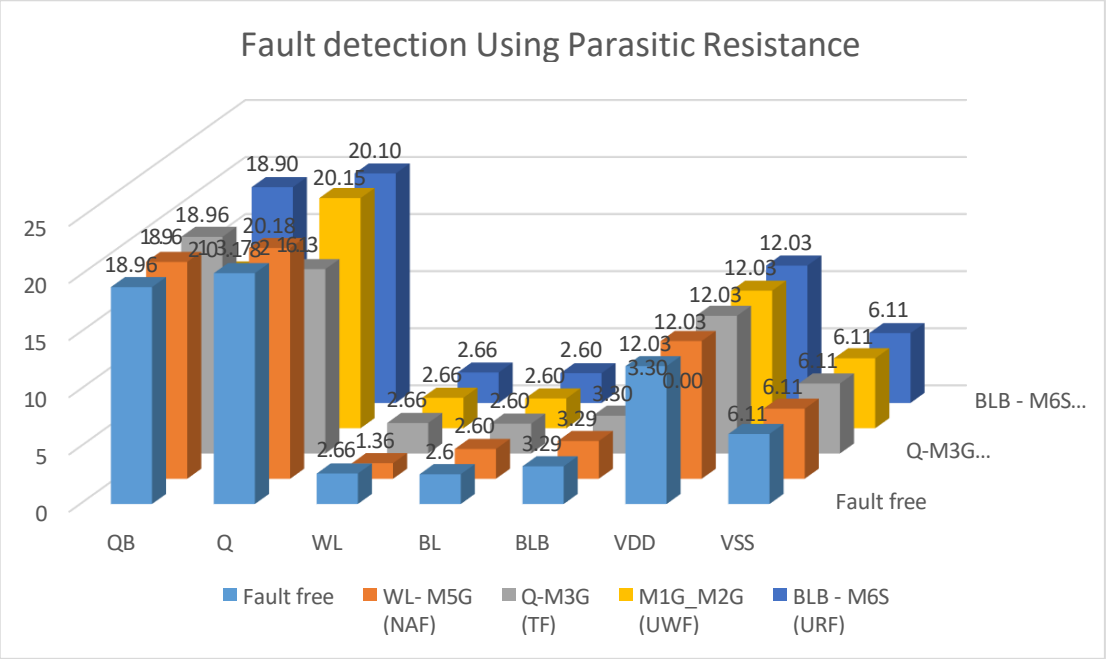


Fig 3.10b. Variation in Resistance for different faults

The figures 3.10a, 3.10b shows the detection of the open faults using parasitic extraction method. Fig 3.10a shows the variation of the parasitic capacitance for the faults NAF, TF, UWF and URF. Open between WL and M_5G causes the No Access Fault. This fault changes the capacitance at the node WL. For fault free SRAM cell, the capacitance at node WL is 0.88 fF, when open fault occurs the capacitance value changes to 0.26fF. Similarly the fault model Q- M3Gcauses the transition fault, changes the parasitic C value at node Q. for fault free SRAM the Parasitic C, at node Q is 3.3 fF, when open defect occurs the capacitance value changes to 2.8fF. Thus variation in the capacitance indicates the defect at the node. Similarly researcher can detect the other faults also.

Similarly by using variation in the parasitic Resistance, researcher detected the faults at the node. Fig. 3.10b shows the parasitic R, changes for the different open faults occur during the manufacturing time.

Table-3.9: Extracted R, C Values Single 6T-SRAM cell for all open defects using 45nm Technology

S.No	Open Defect	Node QB		Node Q		Node WL		Node BL		Node BLB		Node VDD		Node VSS	
		Fault Free C = 2.9fF, R=18.96KΩ		Fault Free C = 3.3fF, R=20.18 KΩ		Fault Free C = 0.88fF, R= 2.66KΩ		Fault Free C = 1fF, R= 2.6KΩ		Fault Free C = 0.61fF, R= 3.29KΩ		Fault Free C = 2.4fF, R=12.032KΩ		Fault Free C = 1.9fF, R= 6.11KΩ	
1	BL-M5S	2.90	18.97	3.30	20.18	0.88	2.66	NA	NA	0.61	3.29	2.4	12.03	1.9	6.11
2	WL- M5G	2.90	18.96	3.30	20.18	0.26	1.36	1.00	2.60	0.61	3.29	2.4	12.03	1.9	6.11
3	WL- M6G	2.80	18.90	3.10	20.12	0.80	1.36	1.00	2.60	0.60	3.30	2.4	12.03	1.9	6.11
4	Q-M1D	2.90	18.96	2.40	12.26	0.88	2.66	1.00	2.60	0.61	3.29	2.4	12.03	1.9	6.11
5	Q-M2D	2.90	18.96	2.70	16.74	0.88	2.66	1.00	2.60	0.61	3.29	2.4	12.03	1.9	6.11
6	Q-M1DM2D	2.90	18.95	1.20	3.62	0.88	2.66	1.00	2.60	0.64	3.32	2.4	12.03	1.9	6.11
7	Q-M3G	2.90	18.96	2.80	16.13	0.88	2.66	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11
8	Q- M4G	2.90	18.96	3.00	18.94	0.88	2.66	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11
9	Q-M3GM4G	2.90	18.94	2.60	14.91	0.88	2.66	1.00	2.60	0.63	3.30	2.4	12.03	1.9	6.11
10	VDD-M1S	2.90	18.94	3.30	20.18	0.88	2.66	1.00	2.60	0.63	3.30	2.1	8.65	1.9	6.11
11	VDD-M3S	2.90	18.92	3.30	20.18	0.88	2.66	1.00	2.60	0.64	3.32	2.1	8.65	1.9	6.11
12	VDD-M1SM3S	2.90	18.94	3.20	20.15	0.88	2.66	1.00	2.60	0.64	3.32	1.9	5.27	1.9	6.11
13	VSS-M2S	2.90	18.96	3.20	20.16	0.88	2.66	1.00	2.60	0.65	3.34	2.4	12.03	1.7	3.93
14	VSS-M4S	2.90	18.96	3.30	20.21	0.88	2.66	1.00	2.60	0.64	3.32	2.4	12.03	1.7	3.93
15	VSS-M2SM4S	2.90	18.96	3.20	20.12	0.88	2.66	1.00	2.60	0.63	3.30	2.4	12.03	1.6	1.76
16	QB - M3D	2.10	11.06	3.30	20.17	0.88	2.66	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11
17	QB - M4D	2.40	15.51	3.20	20.16	0.88	2.66	1.00	2.60	0.64	3.32	2.4	12.03	1.9	6.11
18	QB_M3DM4D	0.52	2.31	3.20	20.16	0.88	2.66	1.00	2.60	0.64	3.32	2.4	12.03	1.9	6.11
19	QB_M1G	2.50	14.94	3.30	20.17	0.88	2.66	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11
20	QB_M2G	2.70	17.74	3.20	20.13	0.88	2.66	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11
21	QB_M1GM2G	2.70	17.74	3.20	20.13	0.88	2.66	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11
22	M1G_M2G	2.30	13.72	3.20	20.15	0.88	2.66	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11
23	M3G_M4G	2.90	18.94	2.60	14.91	0.88	2.66	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11
24	BLB - M6S	2.80	18.90	3.10	20.10	0.88	2.66	1.00	2.60	NA	NA	2.4	12.03	1.9	6.11
25	WL-M5GM6G	2.80	18.90	3.10	20.10	NA	NA	1.00	2.60	0.61	3.30	2.4	12.03	1.9	6.11

3.6 6T SRAM Cell analysis for short Defects

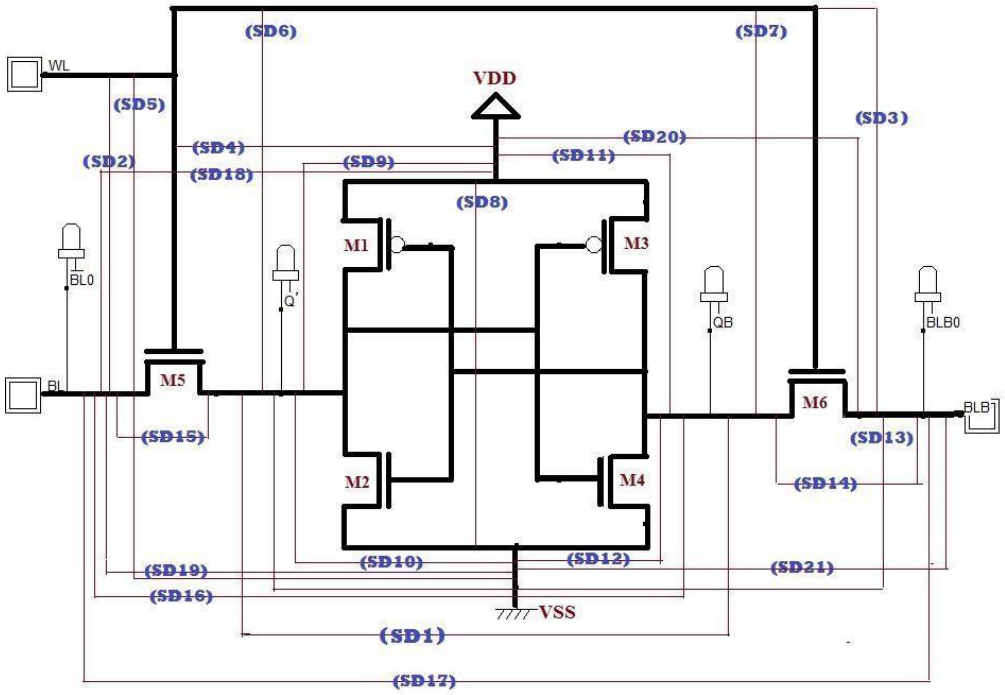


Fig3.11.ProposedFault Model for short Defects

Fig.3.11 shows 6T SRAM cell with seven main nodes BL, BLB, Q, QB, WL, VDD and VSS. Out of which Q, QB are internal nodes through them the cell state can be monitored and WL, BL and BLB are external nodes through these writing and reading operations can be performed. Two more nodes are supply and ground nodes VDD and VSS respectively. All possible shorts between internal and external nodes are considered in the proposed fault model.

Table-3.10**:** Node equivalence corresponding to main nodes.

S. No	Main Node	Node Equivalence
1	Q	DM2,DM1,DM5,GM3,GM4
2	QB	DM3,DM4,SM6,GM2,
3	WL	GM5,GM6
4	BL	SM5
5	BLB	DM6
6	VDD	SM2,SM3
7	VSS	SM1,SM4

Internal node Q is common point to drain of M_1D, M_2D and M_5D transistors. It is also common point for gate of M_3G and M_4G transistors. Hence short between M_1D to QB is equivalent to short between Q and QB. It is true with other equivalent nodes. Including equivalent nodes, all possible short defects between the internal and external nodes are 259. However, excluding equivalent nodes, the actual short defects are found are only 21. For simplicity, the short defects are represented with SD (SD ids the abbreviation for Short Defects) are listed in table 3.11.

Table-3.11: Complete Fault Model Dictionary for short defects for different technologies

S.No	Fault Representation	Short between Nodes	Technology		
			45nm	32nm	7nm
1	SD1	Q-QB	USWF,URF	USWF,URF	USWF,URF
2	SD2	WL-BL	WBAF	TF	WBAF, TF
3	SD3	WL-BLB	USF	USRF-1	WBAF,USRF-1
4	SD4	WL-VDD	Error	Error	Error
5	SD5	WL-VSS	Error	Error	Error
6	SD6	WL-Q	SA0,URF	SA0,URF	SA0,URF
7	SD7	WL-QB	SA1,URF	SA1,URF	SA1,URF
8	SD8	VDD-VSS	UWF,URF0	UWF,URF0	UWF,URF0
9	SD9	Q-VDD	UWF0,URF0	URF0,UWF0	URF0,UWF0
10	SD10	Q-VSS	SA0	URF1,UWF1	URF1,UWF1
11	SD11	QB-VDD	IOF	IOF	IOF
12	SD12	QB-VSS	SA1,URF0	TF, URF0	TF, URF0
13	SD13	Q-BLB	URF	URF	URF
14	SD14	QB-BLB	WBAF	USWF0,USRF0	USWF0,USRF0
15	SD15	Q-BL	SA0(WBAF)	WBAF,SA0	SA0
16	SD16	QB-BL	USWF, USRF	USWF,USRF	USWF,USRF
17	SD17	BL-BLB	USWF, USRF	USWF,USRF	USWF,USRF
18	SD18	BL-VDD	Error	Error(NAF)	Error(NAF)
19	SD19	BL-VSS	Error	Error(NAF)	Error(NAF)
20	SD20	BLB-VDD	Error	Error(NAF)	Error(NAF)
21	SD21	BLB-VSS	Error	Error(NAF)	Error(NAF)

3.6.1 Unstabilized Write Fault (USWF):

A cell suffers from USWF, if a write or transition in a write operation causes Continuous transition in the cell.

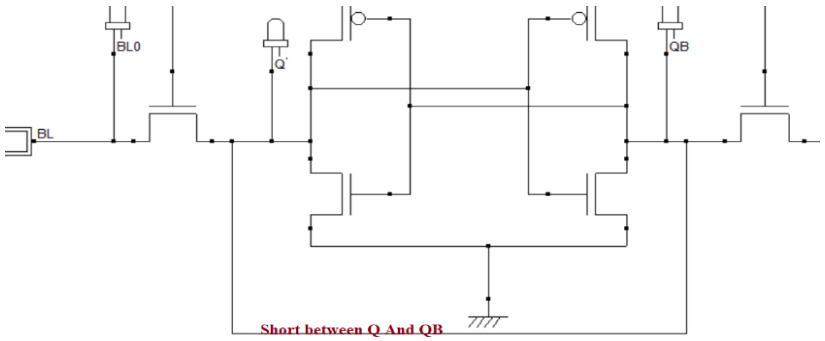


Fig 3.12a. Fault Model for short defect between Q and QB

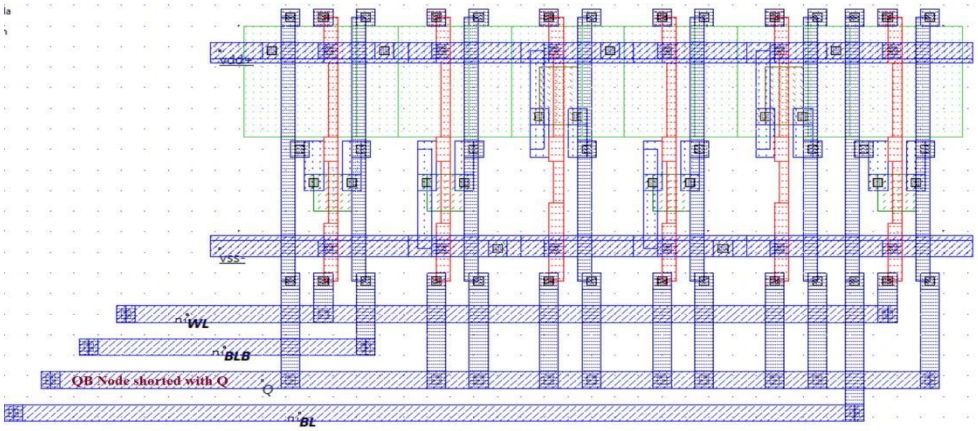


Fig 3.12b. Layout diagram for USWF fault with Short defect at Q-QB

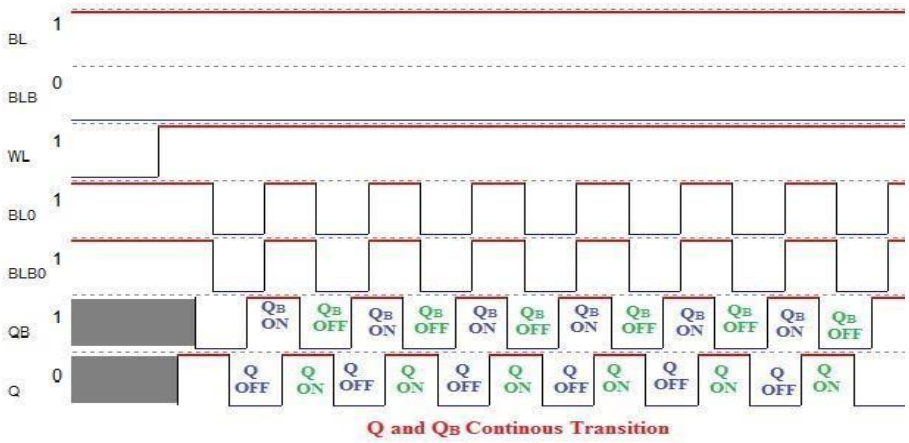


Fig3.12c.Simulation results for Unstabilized Write Fault

As shown in the figure 3.12a when the bit_lines BL and BLB or internal Nodes Q and QB shorted, it causes to ‘Unstabilized Write Fault (USWF)’. The USWF defined as the write operation causes continuous switching between 0 to 1 and 1 to 0. No state is stable. The problem is caused by shorting both bit_lines and internal states together, this will allow the pass transistors to have a common input source that means both the transistors may be at logic 1 or logic 0 position. Because of this common input source, both pass transistors charge and discharge at the same time. This makes the both the states of the SRAM cell either ‘0’ or ‘1’.As seen in Fig 3.12b, because both Q and Qb shorted together the node QB is absorbed by the node Q. As a result, node QB does not exist. This problem manifests itself in the fault models Q-QB, QB-BL, and BL-BLB.

3.6.2 Unstabilized Read Fault (USRF):

If the read operation returns a continuous transition in both the output states and the internal cell data, the cell said to be suffers from ‘Unstabilized Read Fault’. The same reason as in ‘USWF’ can be used to depict this USRF in the read condition.

3.6.3 Write Before Access Fault(WBAF):

If a cell's logic value reaches at the time of write opearation, it is said to have a ‘Write Before Access Fault’. The write operation is completed by placing written data (sensitization) on bit_lines and asserting the WL line. Because to this fault, the sensitized data was written into the cell before asserting the WL line. as in fig. 3.13b

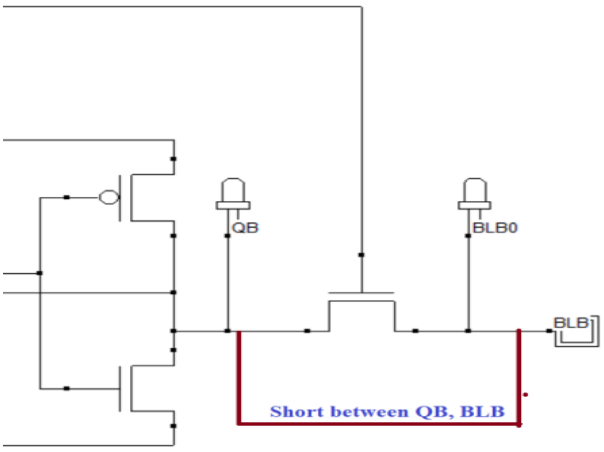


Fig 3.13a. Fault Model for short defect between QB and BLB

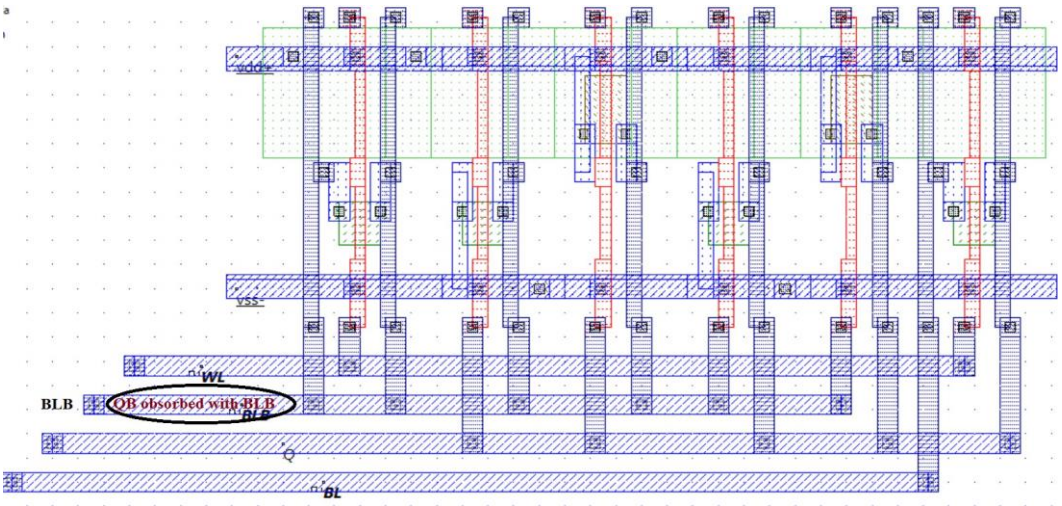


Fig 3.13b. Layout diagram for WBAF fault with Short defect at QB-BLB

In this fault, the WL is always connected to logic 1, as a result, whenever the cell is initialized, the initial data '0' is written into the cell because the write line already in the active high. Similarly, when the logic-1 placed on the bit_line, the data automatically sits in the cell immediately. It is observed that but Q (Q_B) receives data from BL before applying WL. This fault occurs for the fault models WL-QL and QB-BLB

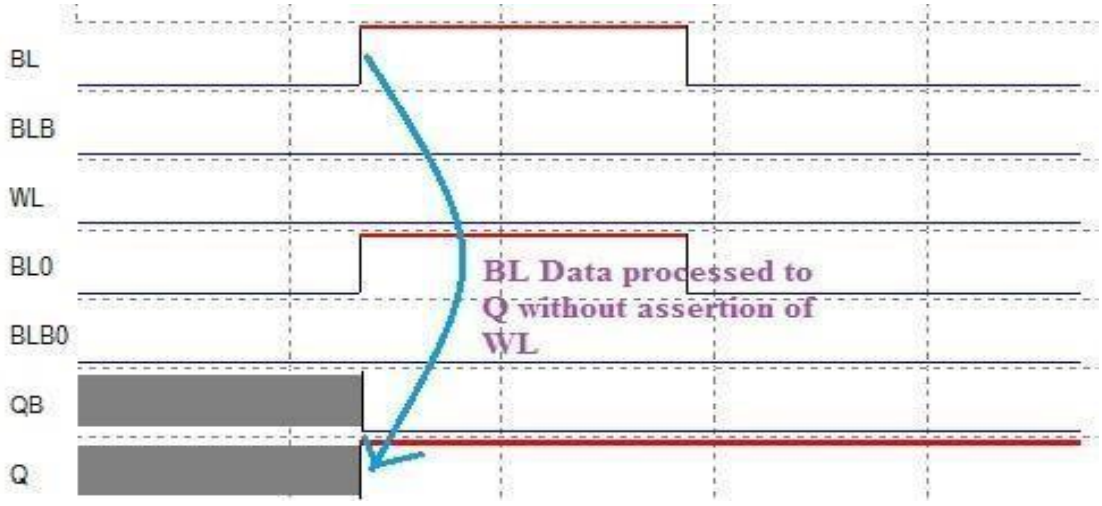


Fig. 3.13cSimulation Results for Write Before Access Fault

3.6.4 Initialization Order Fault (IoF):

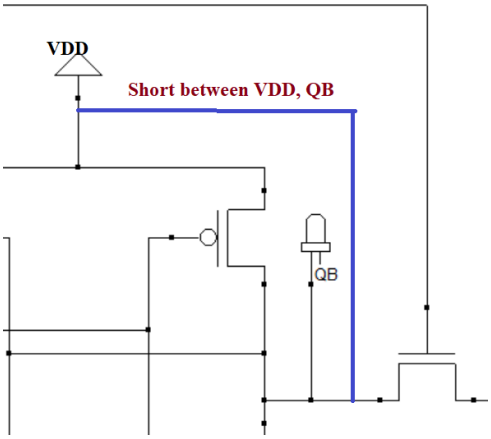


Fig 3.14a. Fault Model for short defect between VDD and QB

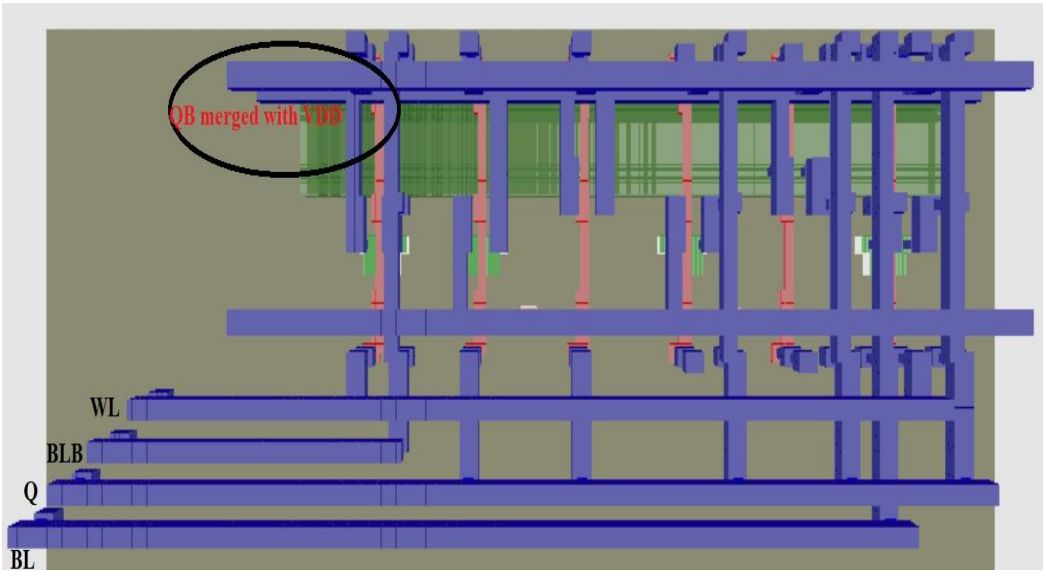


Fig 3.14b. Layout diagram for IOF fault with Short defect at VDD-QB

As shown in fig 3.14c if the operation sequence changes, the initialization problem occurs, resulting in the failure of write/read operations. This fault is represented by a short between QB and VDD, which is analogous to a short between the drain and source of M3 transistors.

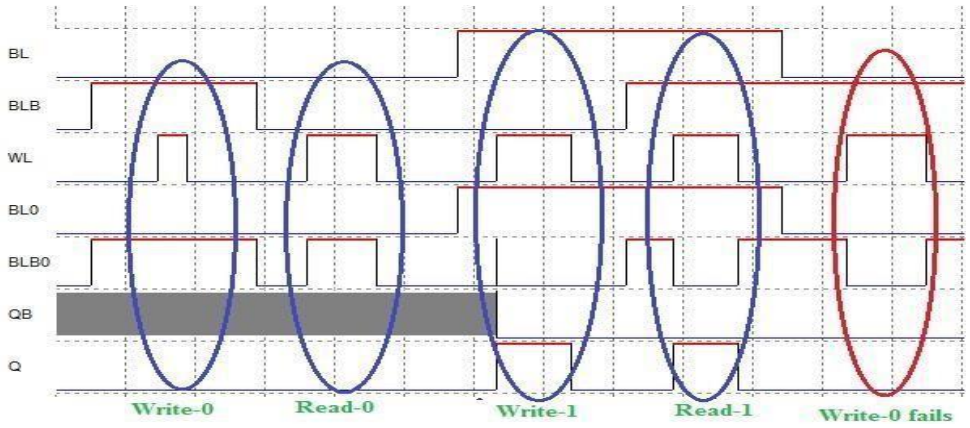


Fig3.14c.Simulation Results for Initialization Order Fault

3.6.5 Read Destructive Retention Fault: this fault occurs, when short defect is between the nodes WL-BLB. In 6T SRAM cell, for write 1 operation, initially we need to set the value of BL=1 and BLB =0, after that WL is asserted, then one will be written into the cell, for read1operation, set BL =1, BLB =1, then WL is asserted, it will force the BL0=1, BLB0 =0. But as shown in fig.3.14 when nodes WL and BLB are shorted, for write 1 operation the cell goes to the undefined state, and for read operation automatically, it stores 1. And when all the bit lines and write lines are set to 0, the cell automatically flips it value that is from logic 1 to logic 0. This is a new type of fault, because no fault primitive defines the above fault; hence it is named as "Read Destructive Retention" fault. Table 3.12. Shows the faults at all possible nodes as shown in the fig.3.11. This experiment carried totally 21 short defect fault models. For each fault model researcher have extracted node resistance and capacitance values, and compared with resistance and capacitance values of fault free SRAM. Table 7. shows the extracted parasitic R and C values for both fault free and fault models at nodes Q, QB, WL, BL, BLB, VDD, VSS.

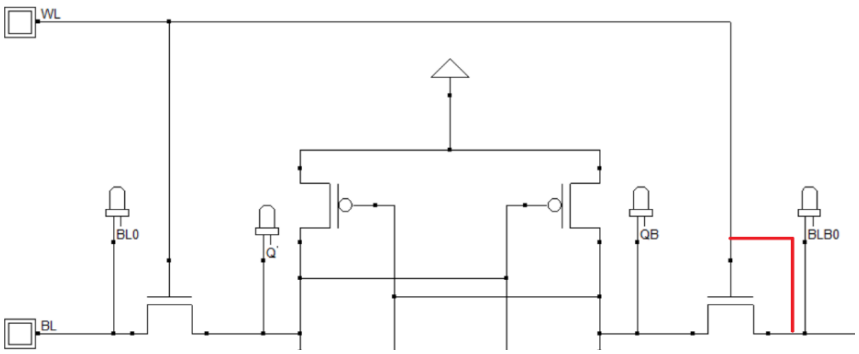


Fig 3.15a. Fault Model for short defect between WL and BLB

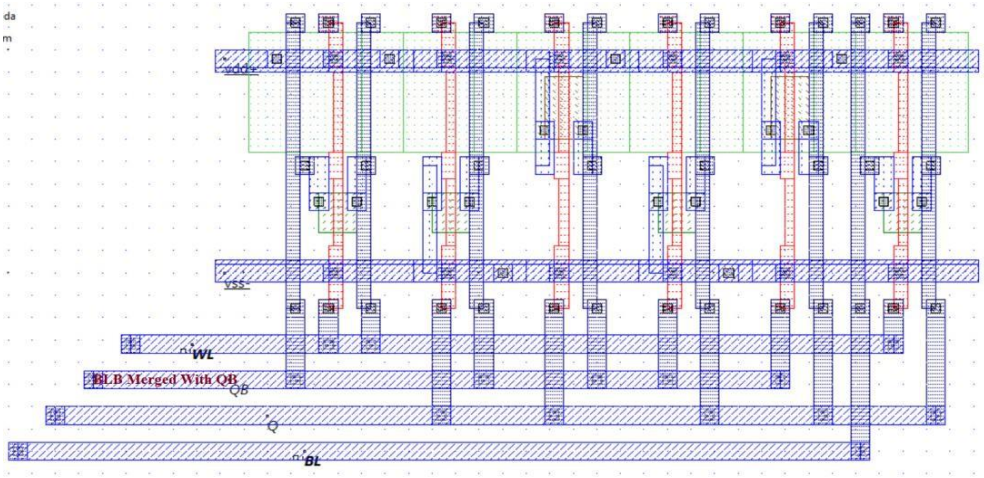


Fig 3.15b. Layout diagram for IOF fault with Short defect at VDD-QB

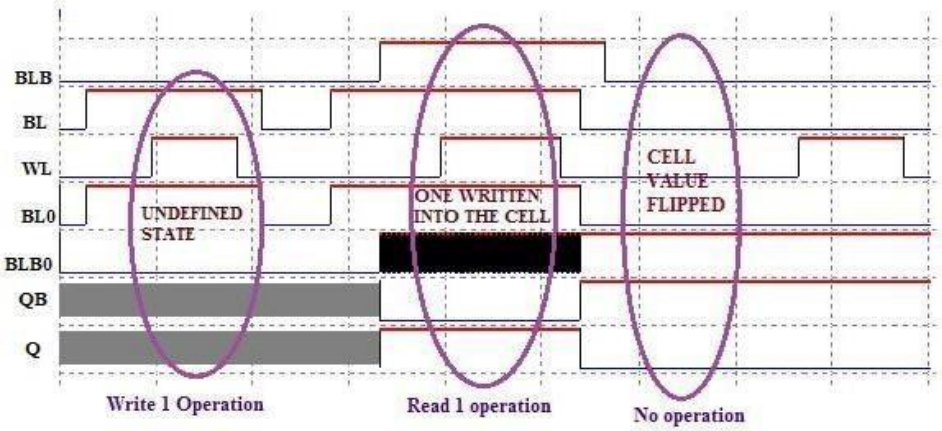


Fig.3.15c. Read Destructive Retention Fault at nodes WL-BLB

It is found that few short defects are exhibiting the same faulty behavior in all three technologies chosen. For example, defect models VDD-VSS represent UWF and URF faults. The UWF Fault occurs with a write operation and the same fault model exhibits URF faults with a read operation. This is due to the fault model VDD being shorted to VSS, then which makes the VDD to the ground potential, hence inverter transistors M_1 and M_3 always stay ON position, leading Q and QB always remain at "0". Hence while writing "1" or writing "0", the node Q and QB will be inactive for accepting new values. For read '0', both BL and BLB results with '0' cause an Undefined Read Fault (URF). The same is true for read operation QB.

Apart from the existing faults, few undetectable faults are identified. For example, defect model WL-BLB for 45nm technology results in **Read Destructive Retention Fault,** however, the same defect model is observed as an Unstabilized Read Fault in 32nm technology, and Write before Access Faults and Unstabilized Read fault for 7nm technology.

Similarly, WL-BL behaves as Stuck at Faults in 45nm, but in the other two technologies, it behaves as Transition faults (TF) and Write Before Access Faults (WBAF). Fault models QB-VSS, WL-BLB follow the same.

3.7 Fault Detection Using Parasitic R, C Method for short faults

3.7.1 Short Fault Detection for 45nm Technology

Table 3.12. Parasitic R, C at affected nodes for chosen short fault models using 45nm Technology

S.No	Open Defect	Node Q		Node QB		Node WL		Node BL		Node BLB		Node VDD		Node VSS	
		Fault Free C = 1900 aF, R=6881 Ω		Fault Free C = 1800 aF, R=7585 Ω		Fault Free C = 663 aF, R= 4712 Ω		Fault Free C = 664.6 aF, R=1216 Ω		Fault Free C = 354 aF, R= 240Ω		Fault Free C = 1900 aF, R=6600Ω		Fault Free C = 1300 aF, R= 2823Ω	
		C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)
1	Q-QB (USWF, URF)	3300	14431	NA	NA	663	4712	647	1215	354	240	1900	6600	1300	2823

2	WL-BL (SA1)	1900	6881	1800	7585	913	5018	NA	NA	354	240	1900	6600	1300	2823
3	WL-BLB (USF)	1900	6876	1700	7580	934	4942	647	1215	NA	NA	1900	6600	1300	2823
4	WL-VDD (Error)	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2000	9843	1300	2823
5	QB-VDD (IOF)	1900	6876	NA	NA	663	4712	646	1215	354	240	2600	13593	1300	2823
6	QB-BLB (WBAF)	1900	6876	2000	7810	663	4712	646	1215	NA	NA	1900	6600	1300	2823
7	QB-BL (USWF, USRF)	1900	6881	NA	NA	663	4712	2200	8769	354	240	1900	6600	1300	2823

Fig.3.16a shows the graphical representation used to detect the faults based on variation in the 'parasitic capacitance' for 45nm Technology is shown in Fig.3.16a. for example short between the WL-BLB, causes to '**Read Destructive Retention Fault** . As expected, the 'parasitic capacitances' will change only the node WL, whereas at other nodes remains same as Fault Free (FF) SRAM cell.

Detection of faults using 'Parasitic Extraction Method' is achieved by comparing 'Parasitic R and C values' of fault model, with 'Parasitic R and C values' of fault free model. For example, if there is a short between node WL and BLB, which causes a new fault named as Undefined Short Fault (USF), this fault can detect by extracting R and C values at node WL and BLB. The parasitic C and R values at WL for fault free SRAM is 663 aF and 4712ohms respectively, when fault induced between node WL and BLB, the parasitic C and R values changes to 934aF and 4942 ohms respectively. And node BLB is absorbed. Whereas no change at other nodes.

Consider the fault model (USWF, URF) corresponding to a short defect modeled by a short between Q and QB, in which the 'parasitic capacitance' variation is more pronounced at affecting node Q, and Node QB is absorbed, whereas other nodes remain the same as Fault Free.

For short defect modeled by short between QB and source VDD which causes 'Initialization Order Fault', Because the nodes QB and VDD shoted the node QB is is absorbed and the 'parasitic R,C values' changes only at the node VDD. In the table it is denoted with 'NA' abbreviation for "Node Absorbed'.

Same explanation hold true for other faults (WBAF, Error, USRF) for short between WL-VDD gives the error fault. As shown in the fig 3.16a, 3.16b it absorbs all the nodes except VDD and VSS and the 'parasitic R,C Values' changes at the nodes VDD

and VSS only.. Fault detection with resistance variation at various node points is shown in Fig.3.16b.

The graph is drawn by taking all faults, with each node on X-axis and their parasitic resistances in ohms on Y-axis. The same explanation applies to defect detection via parasitic resistance variation. As a result, at least one node with a problem will be indicated with its corresponding parasitic values, and the same can be easily discovered during the testing with parasitic values.

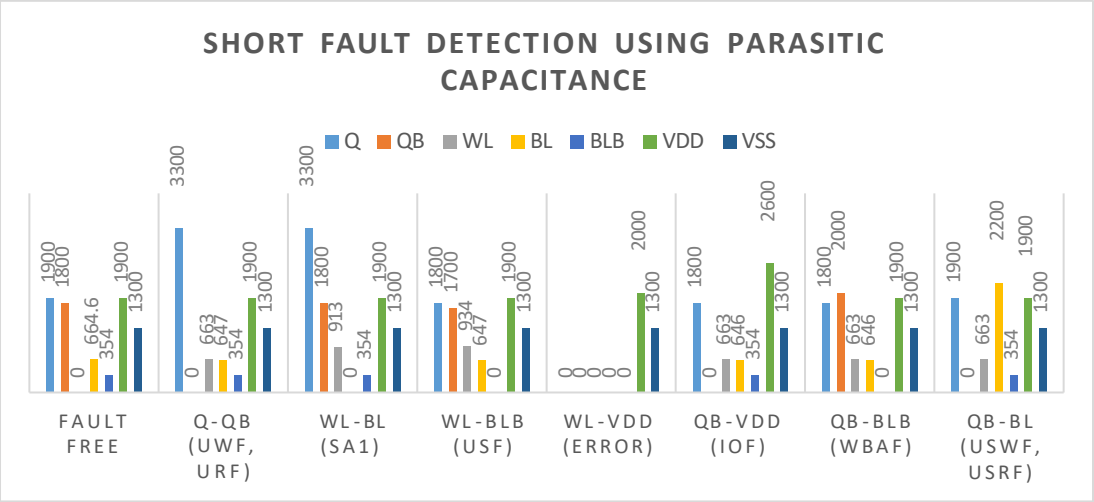


Fig 3.16a. Variation in Capacitance for different short faults (45nm)

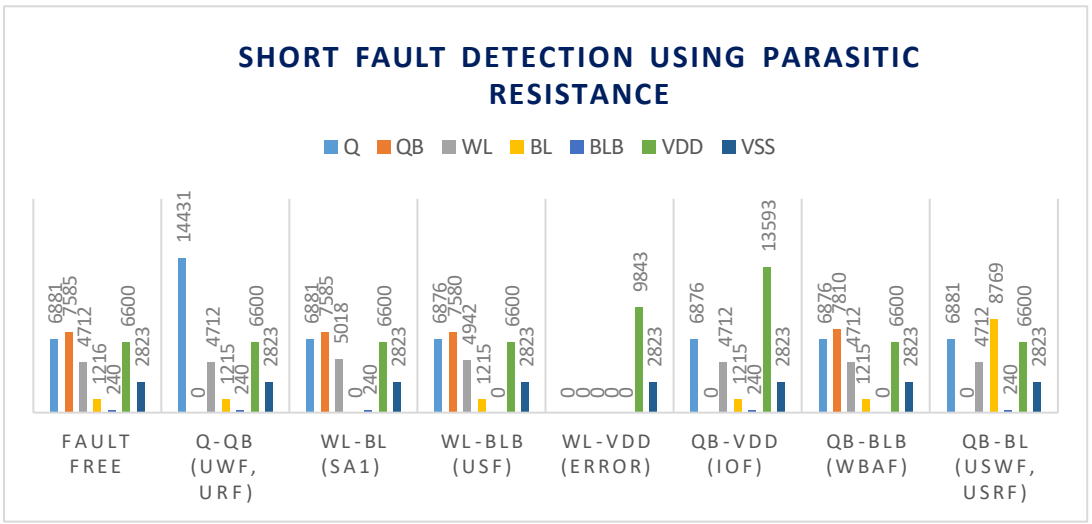


Fig 3.16b. Variation in Resistance for different short faults (45nm)

Table 3.13: Complete fault model dictionary for all short faults using 45nm technology

S.No	Open Defect	Node Q		Node QB		Node WL		Node BL		Node BLB		Node VDD		Node VSS	
		Fault Free C = 1900 aF, R=6881 Ω		Fault Free C = 1800 aF, R=7585 Ω		Fault Free C = 663 aF, R= 4712 Ω		Fault Free C = 664.6 aF, R=1216 Ω		Fault Free C = 354 aF, R= 240Ω		Fault Free C = 1900 aF, R=6600Ω		Fault Free C = 1300 aF, R= 2823Ω	
		C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω
1	Q-QB	3300	14431	NA	NA	663	4712	647	1215	354	240	1900	6600	1300	2823
2	WL-BL	3300	6881	1800	7585	913	5018	NA	NA	354	240	1900	6600	1300	2823
3	WL-BLB	1800	6876	1700	7580	934	4942	647	1215	NA	NA	1900	6600	1300	2823
4	WL-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2000	9843	1300	2823
5	WL-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1900	6600	1600	6858
6	Q-WL	NA	NA	1800	8081	2000	8760	647	1215	354	240	1900	6600	1300	2823
7	QB-WL	1800	6876	NA	NA	2100	12258	647	1215	354	240	1900	6600	1300	2823
8	VDD-VSS	1900	6881	1800	7585	663	4712	664	1216	354	240	1500	3686	1700	5739
9	Q-VDD	NA	NA	1800	7585	663	4712	646	1215	354	240	2500	12877	1300	2823
10	Q-VSS	NA	NA	1800	7585	663	4712	646	1215	354	240	1900	6600	2400	9574
11	QB-VDD	1800	6876	NA	NA	663	4712	646	1215	354	240	2600	13593	1300	2823
12	QB-VSS	1800	6876	NA	NA	663	4712	646	1215	354	240	1900	6600	2100	10291
13	Q-BLB	2000	7101	1800	7585	663	4712	646	1215	NA	NA	1900	6600	1300	2823
14	QB-BLB	1800	6876	2000	7810	663	4712	646	1215	NA	NA	1900	6600	1300	2823
15	Q-BL	NA	NA	1800	7585	663	4712	2100	7944	354	240	1900	6600	1300	2823
16	QB-BL	1900	6881	NA	NA	663	4712	2200	8769	354	240	1900	6600	1300	2823
17	BL-BLB	1900	6881	1800	7585	663	4712	896	1442	NA	NA	1900	6600	1300	2823
18	BL-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2000	7758	1300	2823
19	BL-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1900	6600	1400	3980
20	BLB-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2000	6808	1300	2823
21	BLB-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1900	6600	1300	3030

The complete fault model dictionary for proposed fault models with parasitic R and C values using 32nm technology for 6T SRAM cell is shown in table 3.13. The complete fault model dictionary gives all fault model parasitic values taken node Q,

QB, WL, BL,BLB, VDD and VSS. The variations are further compared with fault free. At which node the fault is imposed that corresponding node parasitics are affected in particular with high parasitic R, C variation. By considering input/output nodes alone, the parasitic values of few fault models exhibiting their unique parasitic R, C variation at input/output nodes with 45nm technology. The same explanation holds for the 32 nm and 7nm technology also.

3.7.2 Short Fault Detection for 32nm Technology

Table 3.14. Parasitic R, C at affected nodes for chosen short fault models using 32nm Technology

nodes	Short defect fault model											
	Fault Free		WL-BL (WBAF, TF)		VDD-VSS (UWF, URF0)		QB-VDD (IoF)		Q-BL (SA0)		QB-BL (USWF, USRF)	
			Effected Node		Effected Node		Effected Node		Effected Node		Effected Node	
	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)
Q	2.9	433	2.90	433	2.90	433	2.90	407	NA	NA	2.90	407
QB	3.1	1170	3.10	1170	3.10	1170	NA	NA	3.10	1170	NA	NA
WL	1.8	180	NA	NA	1.80	180	1.80	178	1.80	180	1.80	180
BL	1.8	158	1.60	236	1.10	158	1.00	157	2.90	529	3.50	941
BLB	0.783	54	0.783	54	0.783	54	0.753	54	0.783	54	0.783	54
VDD	2.7	2071	2.70	2071	2.40	1670	4.00	2787	2.70	2071	2.70	2071
VSS	1.7	402	1.70	402	2.00	805	1.70	402	1.70	402	1.70	402

Fig.3.18a illustrates the fault detection method based on parasitic capacitance change for 32nm technology. Fault model WBAF, TF is created by a short between WL and BL. As expected, parasitic capacitances at other nodes Q, QB, BLB, VDD, and VSS are the same as fault free except at nodes WL and BL, for this fault model node WL is absorbed represented with NA (Node Absorbed)

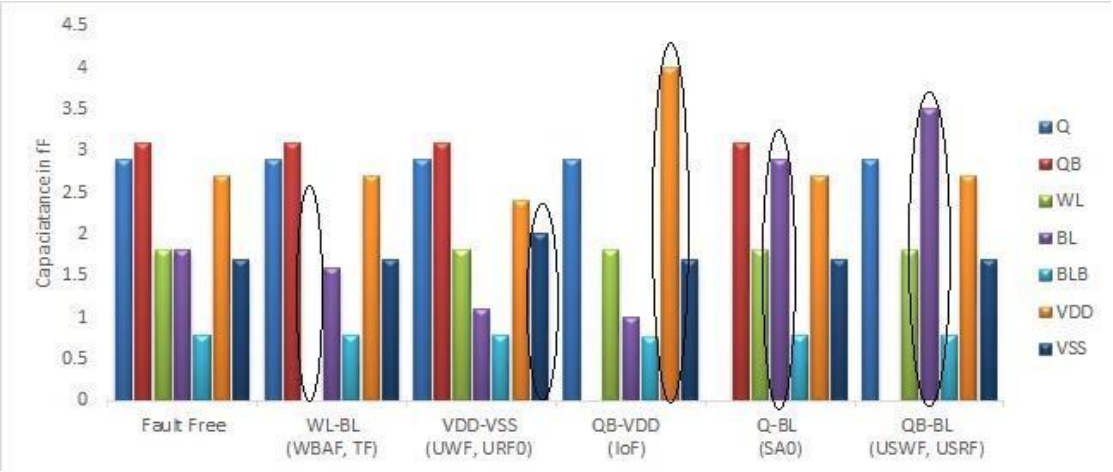


Fig 3.17a Fault detection based on parasitic capacitance variation for short defects

Similar to the fault model UWF, the parasitic capacitance change is more pronounced at impacting nodes VDD and VSS while remaining the same at other nodes that are fault free. URF0 corresponds to a short defect simulated by the short between VDD and VSS. When QB is shorted to VDD to simulate a short defect, parasitic variation is seen at VDD, while node QB is absorbed. For the short defect characterized by Q-BL for fault model SA0, The parasitic variation seen at BL and node Q is absorbed.

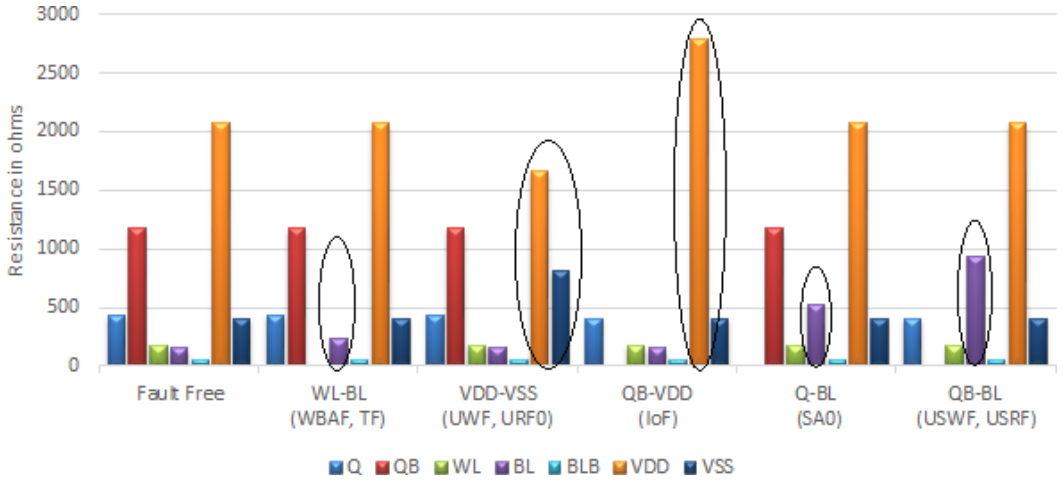


Fig.3.17b Fault detection based on parasitic resistance variation for short faults

Table:3.15: Extracted parasitic R,C values for all Short defects(32nm)

S.No	Open Defect	Node Q		Node QB		Node WL		Node BL		Node BLB		Node VDD		Node VSS	
		Fault Free C=1.8fF,R=677Ω		Fault Free C=1.5fF,R=497Ω		Fault Free C=0.791fF,R=421Ω		Fault Free C=0.701fF,R=75Ω		Fault Free C=0.637fF,R=79Ω		Fault Free C=0.313fF,R=31Ω		Fault Free C=0.313fF,R=13Ω	
		C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω	C in fF	R in Ω
1	Q-QB	2.90	1137	NA	NA	0.75	418	0.67	74	0.59	76	0.31	13	0.31	13
2	WL-BL	1.80	669	1.50	497	NA	NA	0.86	335	0.64	79	0.31	13	0.31	13
3	WL-BLB	1.80	669	1.50	497	1.10	482	0.67	74	NA	NA	0.31	13	0.31	13
4	WL-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.31	13	0.31	13
5	WL-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.31	13	0.31	13
6	Q-WL	2.20	1068	1.50	497	NA	NA	0.67	74	0.59	76	0.31	13	0.31	13
7	QB-WL	1.80	669	NA	NA	1.90	896	0.67	74	0.59	76	0.31	13	0.31	13
8	VDD-VSS	1.80	677	1.50	497	0.79	421	0.70	75	0.64	79	0.31	13	0.31	13
9	Q-VDD	NA	NA	1.50	497	0.79	374	0.671	74	0.59	76	0.31	13	0.31	13
10	Q-VSS	NA	NA	1.50	497	0.79	374	0.671	74	0.59	76	0.31	13	0.31	13
11	QB-VDD	1.60	701	NA	NA	0.74	374	0.671	74	0.59	76	0.31	13	0.31	13
12	QB-VSS	1.60	701	NA	NA	0.74	374	0.671	74	0.59	76	0.31	13	0.31	13
13	Q-BLB	2.10	721	1.50	497	0.79	421	0.671	74	NA	NA	0.31	13	0.31	13
14	QB-BLB	1.80	669	1.80	551	0.79	421	0.671	74	NA	NA	0.31	13	0.31	13
15	Q-BL	NA	NA	1.50	497	0.79	421	1.70	661	0.64	79	0.31	13	0.31	13
16	QB-BL	1.80	669	NA	NA	0.79	421	1.80	549	0.64	79	0.31	13	0.31	13
17	BL-BLB	1.80	669	1.50	497	0.79	421	1.00	138	NA	NA	0.31	13	0.31	13
18	BL-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.31	13	0.31	13
19	BL-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.31	13	0.31	13
20	BLB-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.31	13	0.31	13
21	BLB-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.31	13	0.31	13

3.7.3 Short Fault Detection for 7nm Technology

Table3.16. Variation of parasitic C values for SRAM short defect model (7nm)

Input/output Node	Fault Free	WL-BL (WBAF, TF)	VDD-VSS (UWF, URF0)	QB-VDD (IoF)	Q-BL (SA0)	QB-BL (USWF, USRF)
Q	2.9	2.9	2.9	2.9	NA	2.9
QB	3.1	3.1	3.1	NA	3.1	NA
WL	1.8	NA	1.8	1.8	1.8	1.8
BL	1.8	1.6	1.8	1.8	2.9	3.5
BLB	0.783	0.783	0.783	0.753	0.783	0.783
VDD	2.7	2.7	2.4	4	2.7	2.7
VSS	1.7	1.7	2	1.7	1.7	1.7

Table shows the parasitic Capacitance values for fault free and faulty SRAM cell. We have observed WBAF, TF, UWF, URF, IOF, SAF, USRF and USWF for all fault models for the 7nm technology. When researcher analyze the cell for different technologies, the behavior of the cell also changed. This is shown in the table 11.

Table 3.17. Variation of parasitic Resistance values for SRAM short defect model (7nm)

Node	Fault Free	WL-BL (WBAF, TF)	VDD-VSS (UWF, URF0)	QB-VDD (IoF)	Q-BL (SA0)	QB-BL (USWF, USRF)
Q	433	433	433	407	0	407
QB	1170	1170	1170	0	1170	0
WL	180	0	180	178	180	180
BL	158	236	158	157	529	941
BLB	54	54	54	54	54	54
VDD	2071	2071	1670	2787	2071	2071
VSS	402	402	805	402	402	402

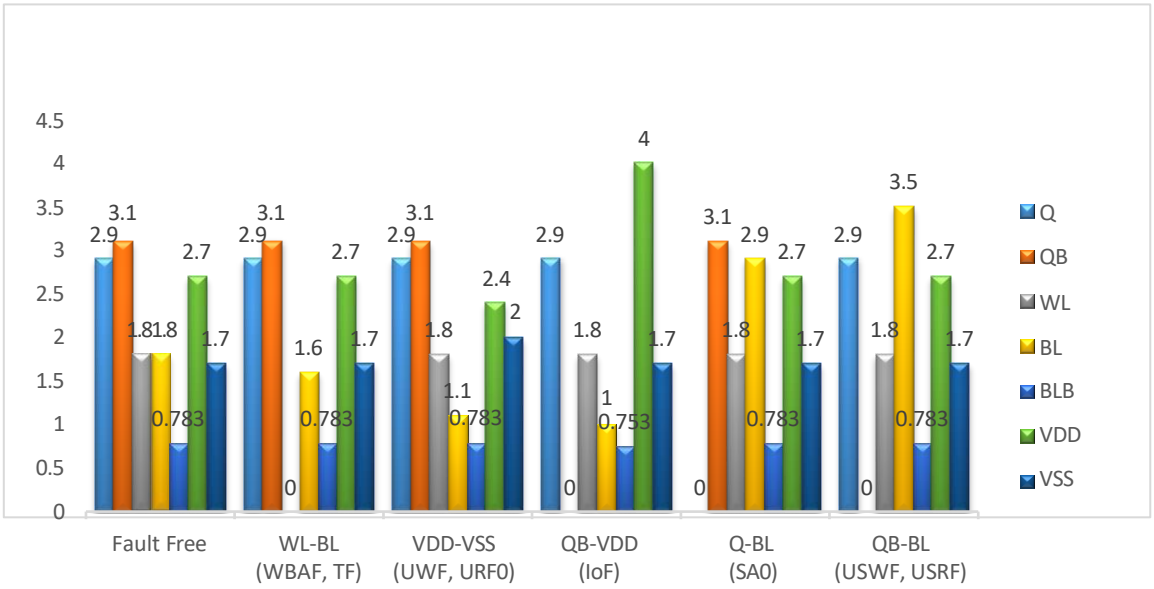


Fig.3.18a Fault detection based on parasitic capacitance variation for short faults (7nm)

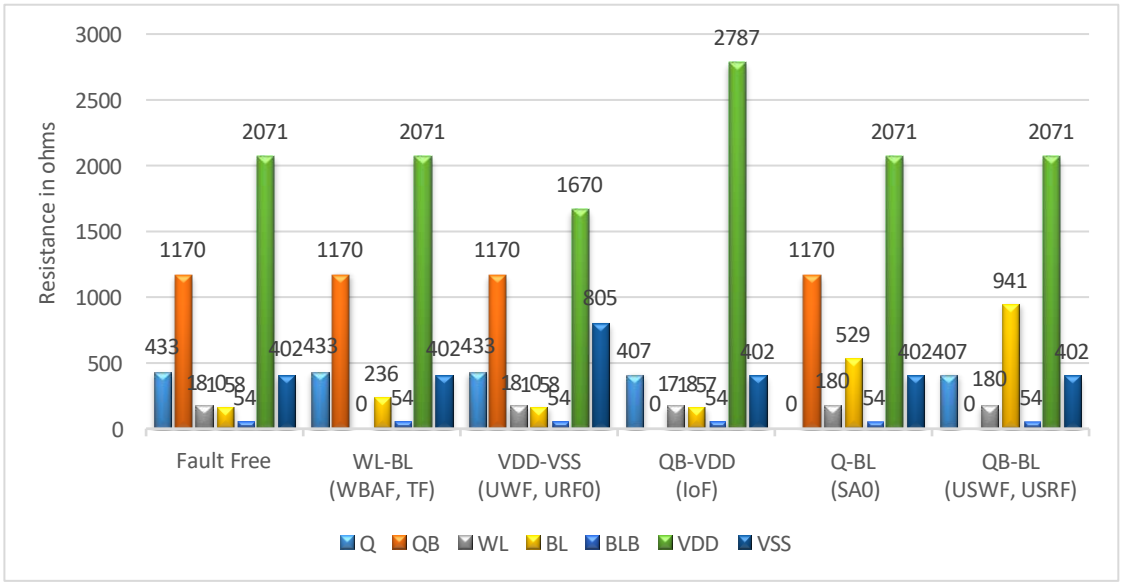


Fig. 3.18b Fault detection based on parasitic capacitance variation for short faults(7nm)

The detection of the faults by using Parasitic R, C values for the 7nm technology is same as fault detection procedure of 32nm and 45 nm technology. That means researcher will extract the parasitic R, C values for fault free and faulty SRAM cell. And compared the extracted R, C values. The changes in the Parasitic R, C values at the node indicates the fault at the node

Table:3.18: Extracted R,C values for all Short defects (7nm)

S.No	Open Defect	Node Q		Node QB		Node WL		Node BL		Node BLB		Node VDD		Node VSS	
		Fault Free C = 2.9fF, R=433Ω		Fault Free C = 3.1fF, R=1170 Ω		Fault Free C = 1.8fF, R= 180Ω		Fault Free C = 1.1fF, R=158Ω		Fault Free C = 0.783fF, R= 54Ω		Fault Free C = 2.7fF, R=2071Ω		Fault Free C = 1.7fF, R= 402Ω	
		C(Ff)	R (Ω)	C(Ff)	R (Ω)	C(Ff)	R (Ω)	C(Ff)	R (Ω)	C(Ff)	R (Ω)	C(Ff)	R (Ω)	C(Ff)	R (Ω)
1	Q-QB	5.50	1583	NA	NA	1.80	178	1.00	157	0.753	53	2.70	2071	1.70	402
2	WL-BL	2.90	433	3.10	1170	NA	NA	1.60	236	0.783	54	2.70	2071	1.70	402
3	WL-BLB	2.90	433	3.10	1170	2.10	219	1.00	159	NA	NA	2.70	2071	1.70	402
4	WL-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.80	2164	1.70	402
5	WL-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.70	2071	2.70	553
6	Q-WL	4.00	565	3.10	803	NA	NA	1.00	157	0.753	54	2.70	2071	1.70	402
7	QB-WL	2.90	433	NA	NA	4.30	1331	1.00	157	0.753	54	2.70	2071	1.70	402
8	VDD-VSS	2.90	433	3.10	1170	1.80	180	1.10	158	0.783	54	2.40	1670	2.00	805
9	Q-VDD	NA	NA	3.00	971	1.80	178	1.000	158	0.783	54	3.60	2409	1.70	402
10	Q-VSS	NA	NA	3.00	971	1.80	178	1.00	158	0.753	53	2.70	2071	3.10	743
11	QB-VDD	2.90	407	NA	NA	1.80	178	1.00	157	0.753	54	4.00	2787	1.70	402
12	QB-VSS	2.90	407	NA	NA	1.80	178	1.00	157	0.753	53	2.70	2071	3.50	1146
13	Q-BLB	3.10	445	3.10	803	1.80	180	1.00	157	NA	NA	2.70	2071	1.70	402
14	QB-BLB	2.90	407	3.40	842	1.80	180	1.00	157	NA	NA	2.70	2071	1.70	402
15	Q-BL	NA	NA	3.10	1170	1.80	180	2.90	529	0.783	54	2.70	2071	1.70	402
16	QB-BL	2.90	407	NA	NA	1.80	180	3.50	941	0.783	54	2.70	2071	1.70	402
17	BL-BLB	2.90	407	3.10	803	1.80	180	1.30	196	NA	NA	2.70	2071	1.70	402
18	BL-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.80	2198	1.70	402
19	BL-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.70	2071	1.80	528
20	BLB-VDD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.80	2101	1.70	402
21	BLB-VSS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.70	2071	1.70	430

3.8 FAULT MODELS IN TWO-CELL SRAM ARCHITECTURE

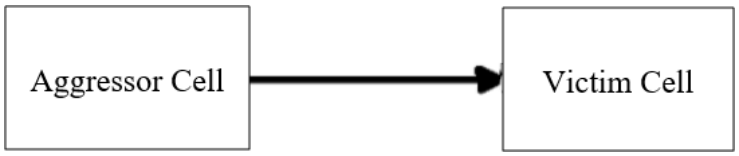
Based on the reviewed literature in this connection, it was noticed that very few studies are available on the coupling faults. The primary reason is, when more than one cell is considered, the fault number would either be doubled or tripled in comparison to single cell faults. Due to this fault exaggeration, using march algorithms, more number of primitive compositions are required, which consumes lot of time. These multi cell faults consider as Linked Faults. There are two types of linked faults i) Coupling faults ii) Neighborhood Pattern Sensitive Fault (NPSF).

3.8.1 Analysis of Coupling Faults:

A cell can develop a coupling fault when it couples with other cells, which causes the cell to malfunction. A cell can be connected with other cells in an exponential number of different ways. The widely used 2-cell coupling fault model makes the assumption that any "two" cells can couple together, which leads to abnormal behavior in these two cells.

There are different types of coupling faults i) State Coupling faults (CFst), ii) Inversion Coupling Fault (CFin) and iii) Idempotent Coupling Fault (CFid).

State Coupling Faults (CFst):



A given value 0 or 1 of the cell in the aggressor word forces a certain value 0 or 1 in a cell of the victim word

The value 0 in aggressor cell causes, victim cell content to be 0

The value 0 in aggressor cell causes victim cell content to be 1

The value 1 in aggressor cell causes victim cell content to be 0

The value 1 in aggressor cell causes victim cell content to be 1

Inversion Coupling Fault (CFin):

A 0 to 1 or 1 to 0 transition write operation in Aggressor Cell, causes the inversion in the victim cell known as Inversion Coupling Fault (CFin) .i.e change in aggressor cell from 0 to1or1 to 0, complements the value in victim cell

Idempotent Coupling Fault (CFid):

A 0 to 1 or 1 to 0 transition write operation in cell of aggressor forces a certain value (0 or 1) in a victim cell

Neighborhood Pattern Sensitive Fault (NPSF):

A Cell i's ability to change influenced by all other memory cell contents, which may be a 0/1 pattern or a transition pattern.

There are three types of NPSFs

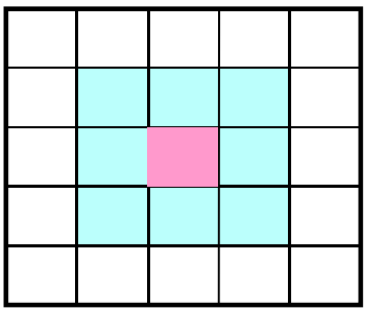
i) Active NPSFs(ANPSF)

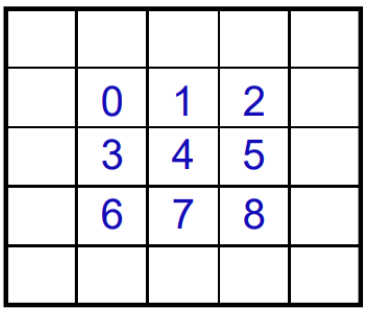
ii) Passive NPSFs(PNPSFs)

iii) Static NPSFs(SNPSF)

Active NPSF: Base cell changes when one deleted neighbourhood cell transitions. The condition to detect and locate active NPSF is each base cell must be read in state 0 and state 1, for all possible deleted neighbourhood pattern changes

Passive NPSF: A certain neighbourhood pattern prevents the base cell from changing Condition for detection and location: – Each base cell must be written and read in state 0 and in state 1, for all deleted neighbourhood pattern changes





Static NPSF: Base cell forced into a particular state when deleted neighborhood contains particular pattern. Condition for detection and location:– Apply all 0 and 1combination stock-cell neighborhood, and verify that each base cell was

3.8.2 Linked Fault Detection using Extraction of Parasitic R, C Method.

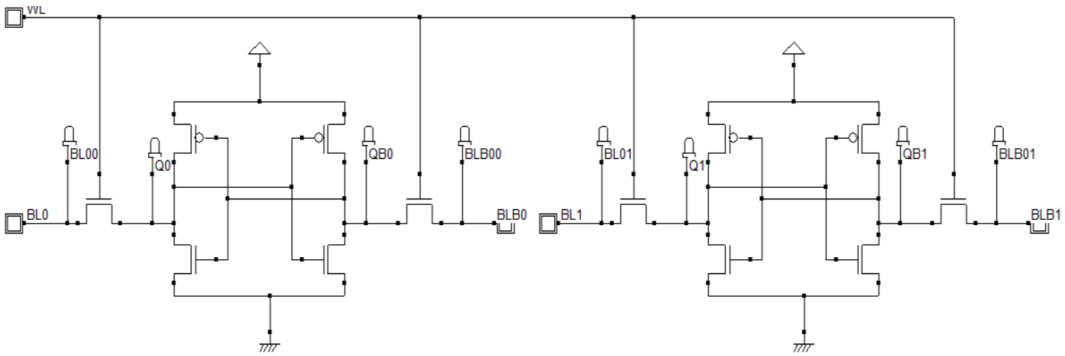


Fig3.19. Two Cell 6T- SRAM with common word line

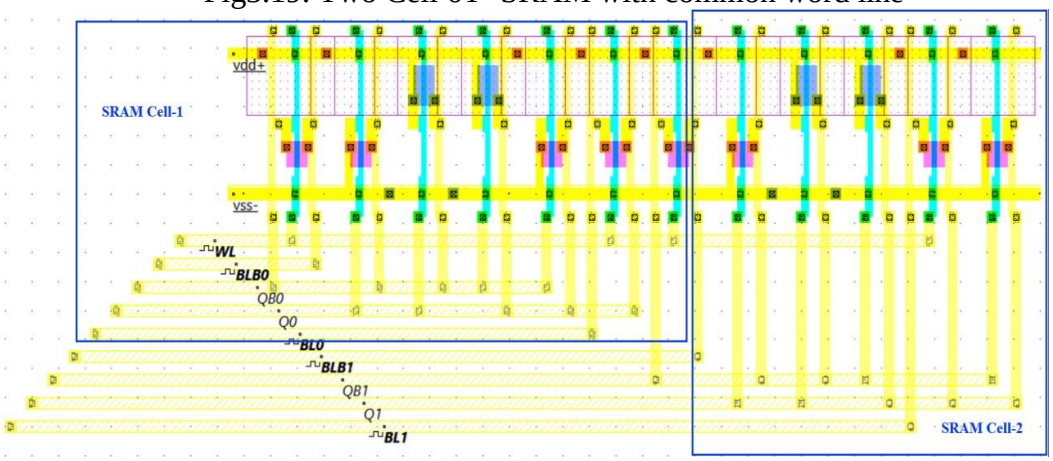


Fig3.20. Layout Diagram for two cell 6T SRAM Cell

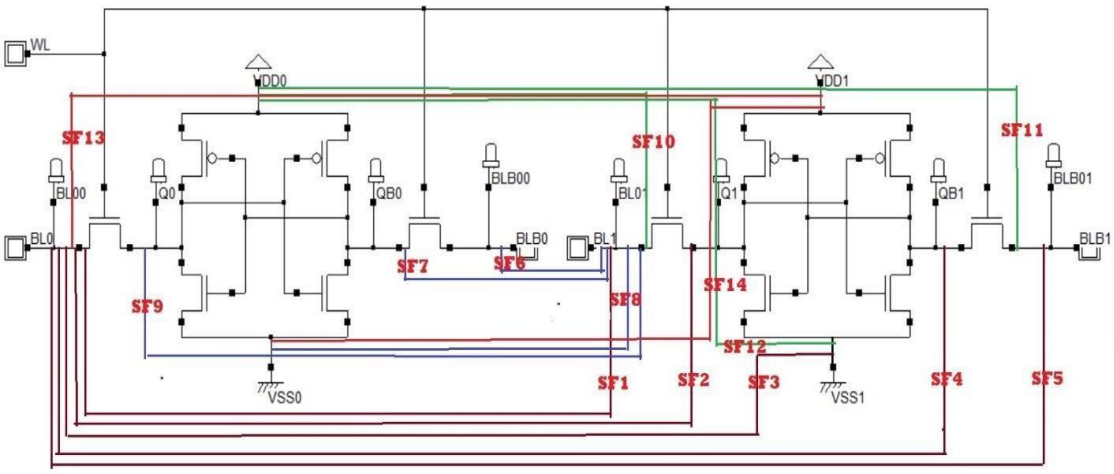


Figure 3.21: Proposed Two Cell SRAM Fault Model for short defects

Fig. 3.21 shows the proposed fault model for two cell SRAM. In the figure the Short Fault abbreviated as SF. The cell-1 acts as an aggressor cell and cell-2 acts as Victim cell. The nodes in the Aggressor Cell represented with Q0, QB0, WL, BL0, BLB0, VDD0 and VSS0, The nodes in the victim Cell represented with Q1, QB1, WL, BL1, BLB1, VDD1 and VSS1. As shown in the figure, the effect of short defects between aggressor cell and victim cell is analyzed. SF1 is a short between the bit line of cell-0 to bit line of cell- 1(BL0-BL1). SF2 represents the short between Cell-0 bit line to Cell-1 internal node Q1 (BL0-Q1).In the proposed fault model there totally 26 possibilities for the short faults.

Table shows the types of faults occurs for the all fault models. For the fault model Q0-BL1, researcher perform all possible write operations 00, 01, 10 and 11, and observed the effect on victim cell.

Table19: Fault Dictionary for two Cell 6T SRAM for coupling short faults

Short at Nodes	Write Operation	Fault Occurred	Fault type
Q0-BL1	00	00	State Coupling Fault (CFst)
	01	00	
	10	11	
	11	11	
Q0-Q1	00	00	State Coupling Fault (CFst)
	01	11	
	10	00	
	11	11	
	00	10	Inversion Couplin Fault(CFin)

	01	01	Fault masking
	10	10	Fault masking
	11	01	Inversion Couplin Fault(CFin)
Q0-BLB10	00	10	Inversion Couplin Fault(CFin)
	01	01	Fault masking
	10	10	Fault masking
	11	01	Inversion Couplin Fault(CFin)
Q0-VDD1	00	UWF	Undefined Write Fault
	01	UWF	Undefined Write Fault
	10	10	Fault masking
	11	11	Fault masking
VDD0-Q1	00	UWF	Undefined Write Fault
	01	01	Fault masking
	10	UWF	Undefined Write Fault
	11	11	Fault masking
VDD0-VSS1	00	UWF,0	Undefined Write Fault, No Fault
	01	UWF,1	Undefined Write Fault, No Fault
	10	UWF,0	Undefined Write Fault, No Fault
	11	UWF,1	Undefined Write Fault, No Fault
VDD0-QB1	00	00	Fault masking
	01	UWF	Undefined Write Fault
	10	UWF	Undefined Write Fault
	11	UWF	Undefined Write Fault
QB0-BL1	00	01	Inversion Couplin Fault(CFin)
	01	01	Fault masking
	10	10	Fault masking
	11	10	Inversion Couplin Fault(CFin)
QB0-Q1	00	10	Inversion Couplin Fault(CFin)
	01	01	Fault masking
	10	10	Fault masking
	11	01	Inversion Couplin Fault(CFin)
QB0-QB10	00	00	Fault masking
	01	11	State Coupling Fault(CFst)
	10	00	State Coupling Fault(CFst)
	11	11	Fault masking
	00	00	Fault masking

QB0-BLB10	01	USWF, 1	Unstabilised Write Fault, No Fault
	10	USWF, 0	Unstabilised Write Fault, No Fault
	11	11	Fault masking
QB0-VDD1	00	00	Fault masking
	01	01	Fault masking
	10	UWF	Undefined Write Fault
	11	UWF	Undefined Write Fault
VSS1-Q0	00	00	Fault masking
	01	01	Fault masking
	10	00	Inversion Couplin Fault(CFin)
	11	01	Inversion Couplin Fault(CFin)
VSS1-QB0	00	10	Inversion Couplin Fault(CFin)
	01	11	Inversion Couplin Fault(CFin)
	10	10	Fault masking
	11	11	Fault masking
VSS0-Q1	00	00	Fault masking
	01	00	Inversion Couplin Fault(CFin)
	10	10	Fault masking
	11	10	Inversion Couplin Fault(CFin)
VSS0-QB1	00	01	Inversion Couplin Fault(CFin)
	01	01	Fault masking
	10	11	Inversion Couplin Fault(CFin)
	11	11	Fault masking
VSS0-VDD1	00	0,UWF	No Fault, Undefined Write Fault
	01	0,UWF	No Fault, Undefined Write Fault
	10	1,UWF	No Fault, Undefined Write Fault
	11	1,UWF	No Fault, Undefined Write Fault
VDD0-BLB1 VSS1-BLB0 VSS1-BL0 BL0-VDD1 VSS0-BL1 VSS0-BLB1 VSS0-WL VDD0-BL1	00	ERROR	No Write and Read operations are possible
	01		
	10		
	11		

State Coupling Fault (CFst):

As shown in the table, researcher have consider the internode Q0 of cell-0, and shorted with all nodes of Cell-1. Q0-BL1 represents the short between the internode Q0 (cell-0) with input bit line BL1(Cell-1). For this short fault, when the researcher will perform the write operation on Cell-0and Cell-1, the researcher have observed that cell-0 value forced to cell-1. For example when the researcher try to write 0 in cell-0

and 1 in cell-1, 0 should be stored in Cell-0 and 1 should be stored in cell-1, but 0 is stored in cell-1,that is cell-0 value is stored in Cell-1. Similarly when the researcher try to write 1 in cell-0 and 0 in cell- 1, 1 should be stored in Cell-0 and 0 should be stored in cell-1, but 1 is stored in Cell-1that is cell-0 value stored in Cell-1. Same has been observed for all four combinations 00,01,10, and 11.This type fault is called as State Coupling Fault(CFst). This fault occurs at the nodes Q0- BL1, Q0-Q1, QB0 – QB10.

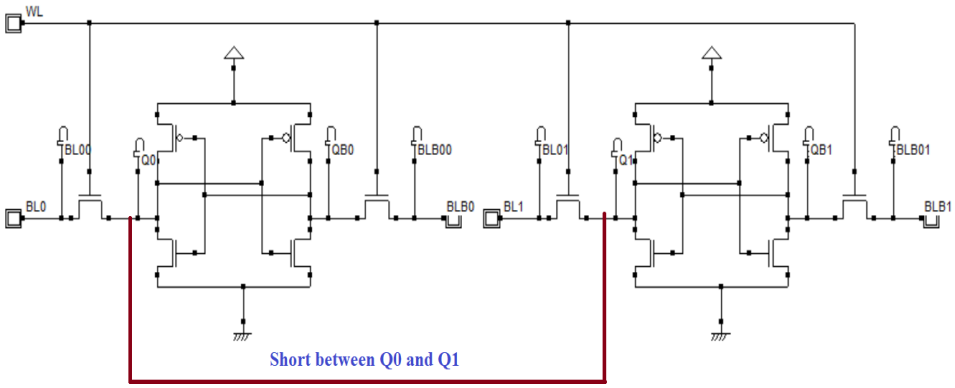


Fig 3.22(a). Fault Model for short defect between Q0 and Q1

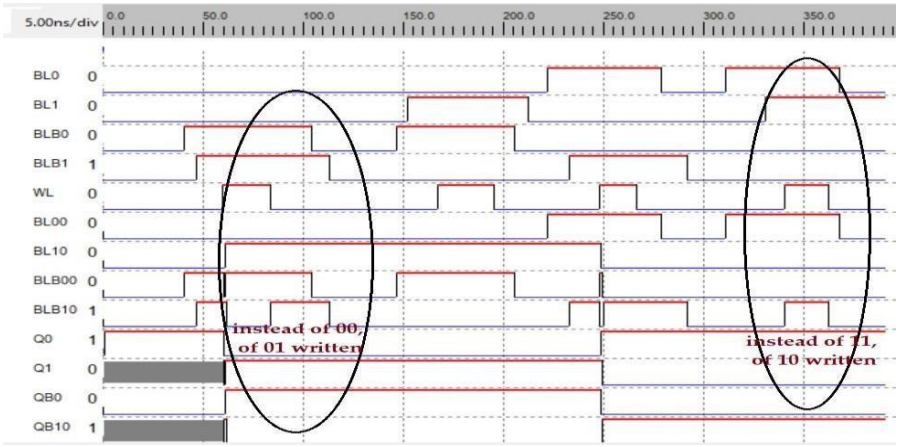


Figure 3.22(b) : Simulation results for State Coupling Faults

Inversion Coupling Fault (CFin):

The switch in write operation performed in ‘aggressor cell’, causes the inversion in the ‘victim cell’**.**

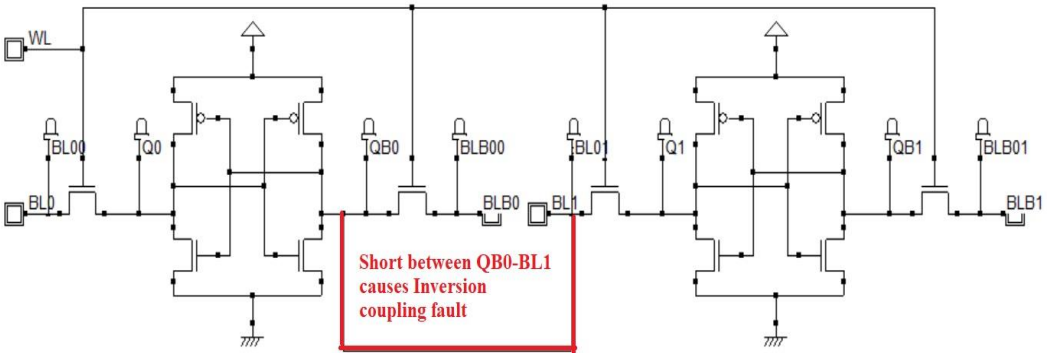


Figure 3.23(a): QB0 and BL1 short defect Fault Model

As shown in the figure 3.4 the QB of the aggressor cell (QB_0), shorted with bilt_line of the victim cell (BL_{01}), this causes inversion coupling Fault.

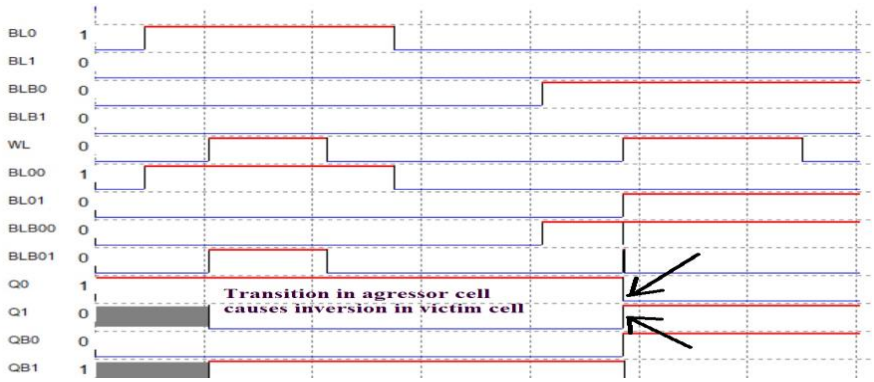


Fig.3.23(b). Simulation results for inversion Coupling Faults

Figure 3.5 shows the simulation results for the short defect QB0 –BL01. As shown in the figure the inversion in the aggressor cell from logic-1 t o logic-0, causes the inversion in the victim cell from logic-0 to logic-1. This type fault is known as inversion coupling fault.

In the above example read and write operations in aggressor cell performed and observed the effect on victim cell. In other way the researcher have performed the read and write operations in both the cells simultaneously and observed the effect of aggressor cell on victim cell. In the second case when the researcher shorted QB0 with BL1, it is observed that when the researcher try to write 0 in cell-0 and 0 in cell-1, 0 should be stored in Cell-0 and 1 should be stored in cell-1, but instead of storing 0 in

cell-0, 1 is stored in the cell-0. Similarly when the researcher try to write 1 in cell-0 and 1 in cell-1, instead of storing 1 in cell-0, 0 is stored in the cell-0. This type of fault is known as Inversion coupling fault .(CFin). This fault Occurs at the nodes Q0-QB10,Q0-BLB10, QB0-BL1, QB0- Q1, VSS1-Q0, VSS1-QB0, VSS0-Q1, VSS0-QB1

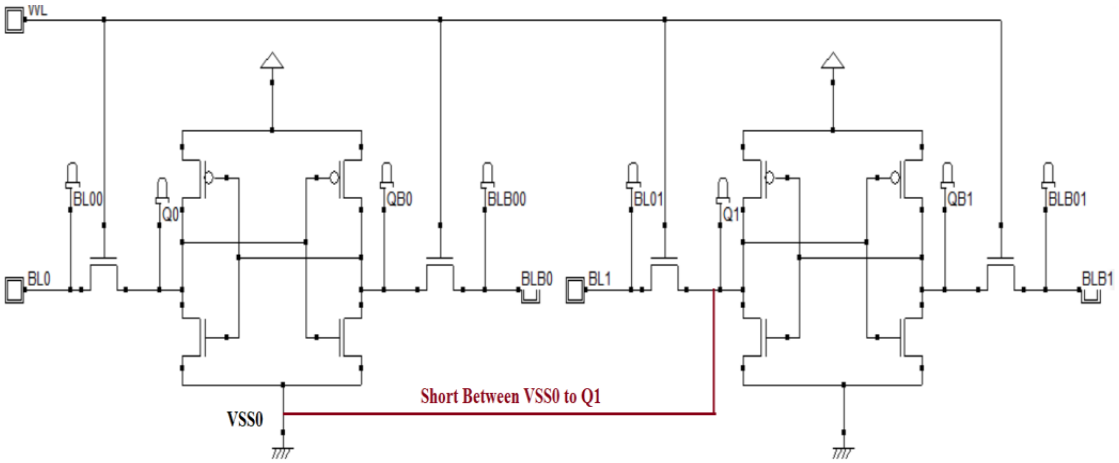


Fig 3.24(a). Fault Model for short defect between VSS0 and Q1

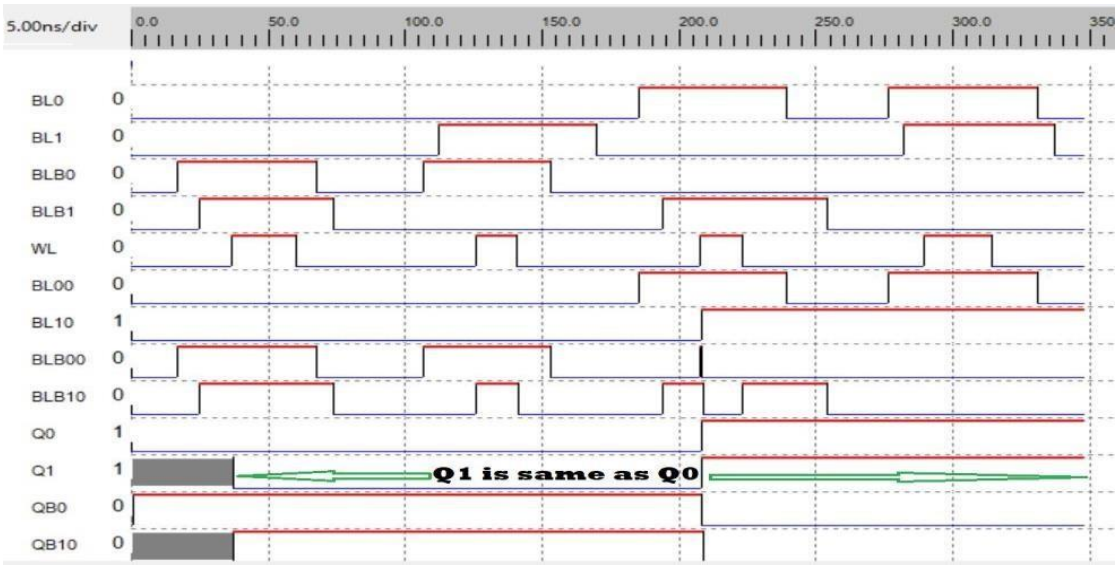


Fig. 3.24(b). Simulation results for inversion Coupling Faults

Fault Masking:

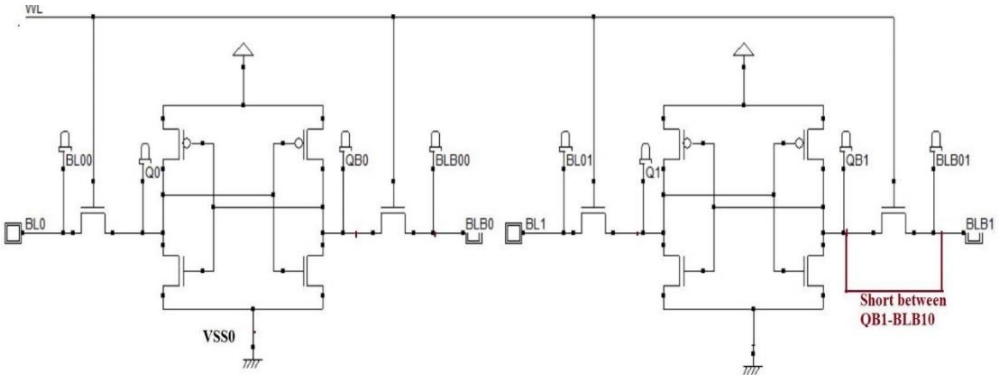


Fig 3.25(a). Fault Model for short defect between QB1 and BLB10.

when the researcher shorted QB1 with BLB1, As shown in the figure 3.24, for write ‘1 operation assign ‘1’ to the bit lines BL0 and BL1, and assign ‘0’ to the bit lines BLB0 and BLB1, then ascertain the word line, it is observed that logic ‘1’ is stored in both the cells. For write ‘0’ operation, assign ‘0’ to the bit lines BL0 and BL1, and assign ‘1’ to the bit lines BLB0 and BLB1, then ascertain the word line, it is observed that logic ‘0’ is stored in both the cells. Even if the researcher have the short defect between nodes QB1 and BLB01, the functional operation the cells are same as the fault free cell. This type of faults known as Fault Masking.. This is occurs at the nodes Q0-QB1, Q0-BLB10, Q0-VDD1, VDD0-Q1, VDD0-QB1, QB0-BL1, QB0-Q1, QB0-QB10, QB0-BLB10, QB0- VDD1, VSS1-Q0, VSS1-QB0, VSS0-Q1, VSS0-QB1

Figure 3.25 b: Simulation results for short fault QB1-BLB10

Undefined Write Fault:

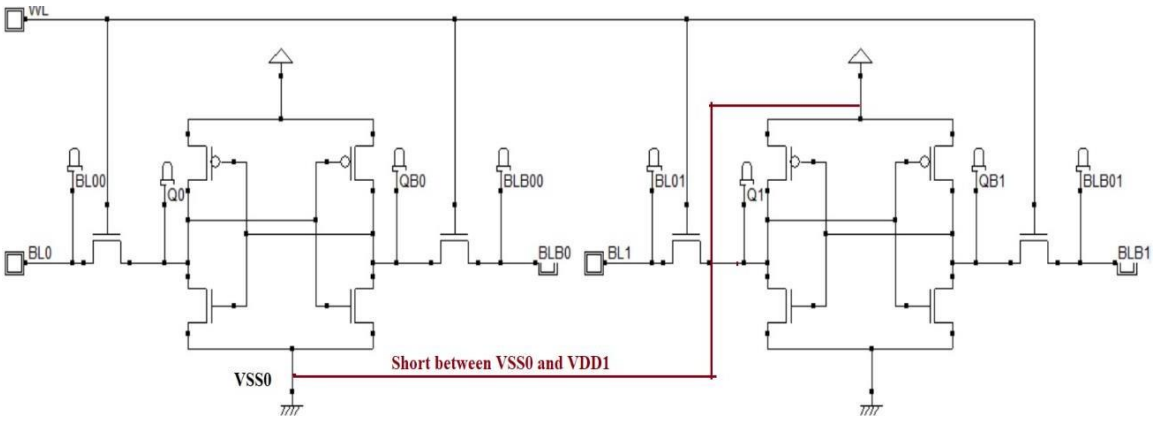


Fig 3.26(a). Fault Model for short defect between VSS0 and Q1

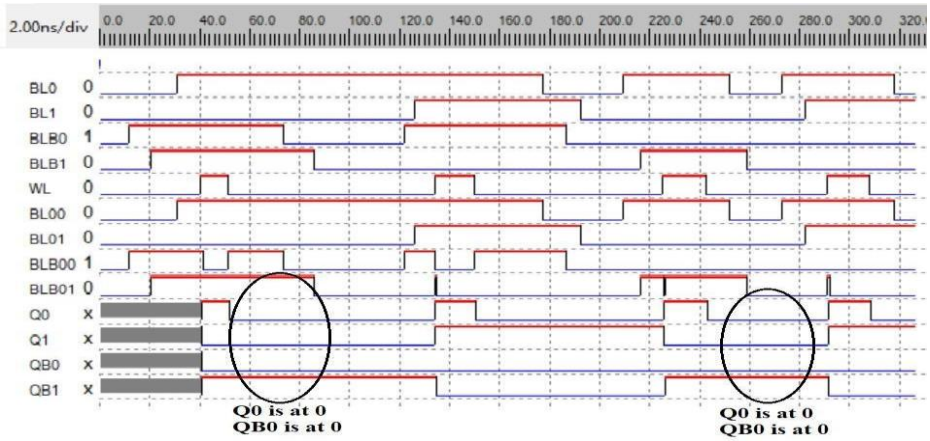


Fig.3.26 (b). Simulation results for Undefined Write Fault

The memory cell said to have Undefined Write Fault, if the cell is brought to in an undefined state through a write operation. Undefined means, the cell state goes to neither '1' nor '0' with a read operation, This fault occurs at the nodes Q0-VDD1, VDD0-Q1, VDD0- VSS1, VDD0-QB1, QB0-VDD1, VSS0-VDD1

Unstabilised Write Fault:

A cell suffers from USWF, if an operation of writing or transitioning in a write operation generates a continuous transition within the cell. A large-sized complicated SRAM circuit has additional bit_lines or column_lines, with the possibility to come closure. Writing '1' or '0' causes the cell to continuously change between these states, with neither being stable. The fault is created by shorting both Q0 and QB0 together

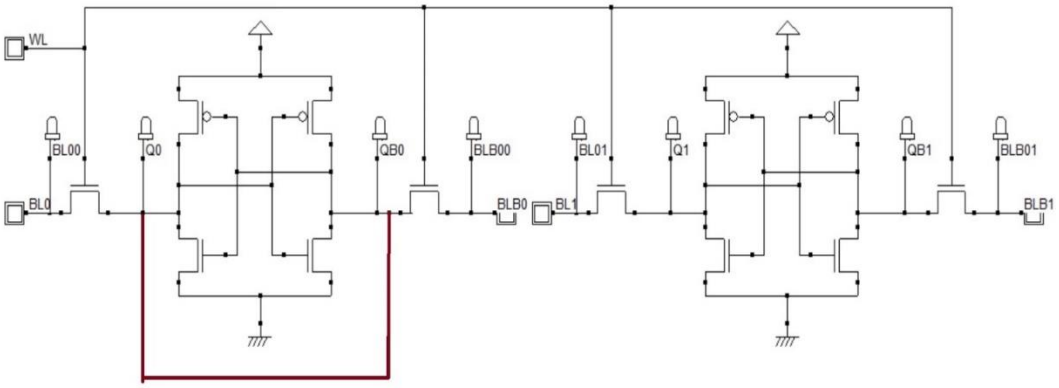


Fig 3.27(a). Fault Model for short defect between Q0 and QB0

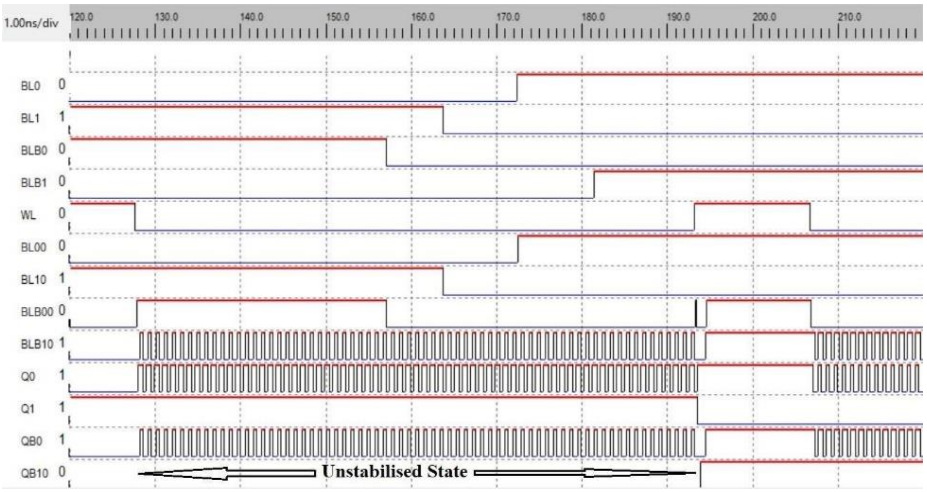


Fig.3.27 (b). Simulation results for USWF

3.1.4 Inaccessible State Coupling Fault (CFist):

State Coupling Fault (CFst)

Two memory cells have a 'state coupling fault', if coupled cell is forced to a certain value 'x' (could be either '0' or '1') if coupling cell is given a state. In contrary CF_{in}, CF_{id}, state fault does not stimulated by transition in write operation, but due to some connection between two cells CFst arise. In other words CFst in cell-2 is caused by a state of the cell-1 rather than by the transition of the cell-1.

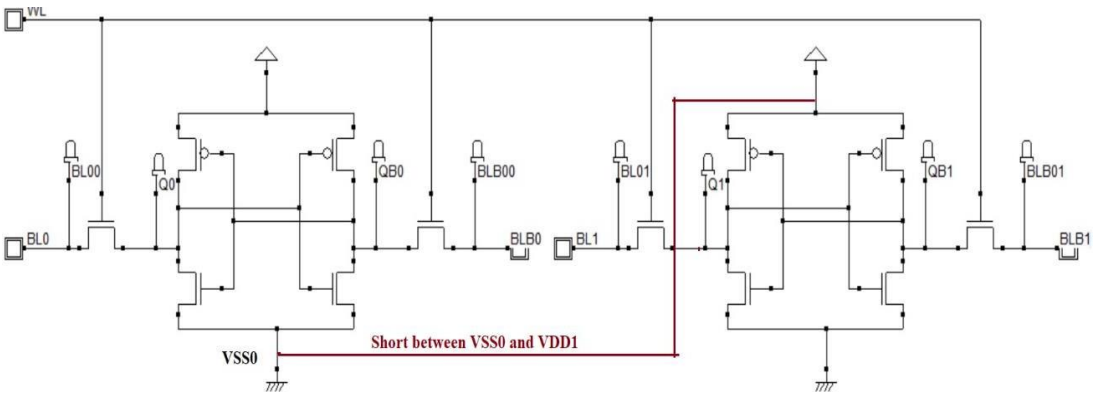


Figure 3.28a: Fault Model for short defect between VSS0 and VDD1

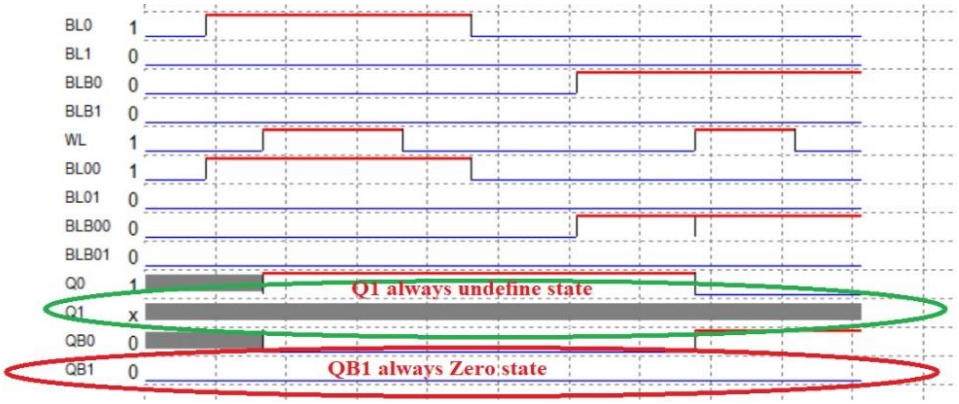


Figure 3.28b: Simulation results for short fault VSS0-VDD1

As shown figure 3.28a when the researcher have shorted the nodes VSS0 and VDD1, it causes a new fault which are not defined by fault primitives.

- In this case transition in the aggressor cell does not effect on the victim cell but always the present state Q1 goes to undefined state and QB1 forces to always zero this operation is ambitious which is not detected by the any existing method.
- As QB1 always forces to a fixed state zero, before performing any operation, this is similar to state coupling fault.
- As Q1 always in the undefined state hence named it as Inaccessible State Coupling Fault and it is Denoted as CF_{ist}
- The subscript ist represents the inaccessible state coupling fault.

Table.3.20 Fault Detection with Parasitic R, C in Two Cell Fault Models:

Nodes	Fault Free		Q0-BL1 State Coupling Fault (CFst)		QB0-BL1 Inversion Couplin Fault (CFin)		Q0-VDD1 UWF		S1-BL0 NAF	
	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)	C(fF)	R(Ω)
Node Q0	7.9	1277	NA	NA	7.90	1307.00	NA	NA	NA	NA
Node QB0	7.7	1194	7.80	1213.00	NA	NA	7.80	1213.00	NA	NA
Node WL	8.2	677	8.20	677.00	8.20	677.00	8.20	677.00	NA	NA
Node BL0	1.1	147	1.1	147	1.10	147.00	1.10	147.00	NA	NA
Node BLB0	1.2	241	1.2	241	1.20	241.00	1.20	241.00	NA	NA
VDD0	1.9	7	1.9	7	1.9	7	1.9	7	1.9	7
VSS0	1.9	7	1.9	7	1.9	7	1.9	7	1.9	7
Node Q1	9.2	1297	9.2	1314	9.2	1314	9	1314	NA	NA
Node QB1	9.1	1202	9	1219	9	1219	9	1219	NA	NA
Node BL1	1.7	150	8.90	1452.00	8.70	1357.00	1.70	150.00	NA	NA
Node BLB1	1.8	245	1.80	244.00	1.80	244.00	1.80	245.00	NA	NA

Detection of faults using parasitic extraction method is achieved by comparing parasitic R and C values of fault model, with parasitic R and C values of fault free model. For example, if there is a short between node Q0 of Cell-0 and Bit-line BL1 of Cell-1, which causes State Coupling fault (CFst), this can detect by extracting R and C values at node Q0 of Cell-0and Bit-line BL1 of Cell-1. The parasitic C and R values at Q0 for fault free SRAM is 7.9fFand 1277Ω and for BL1 the capacitance value is 1.7fF and resistance value 150 ohms respectively, when fault induced between node Q0 of Cell-0 and Bit-line BL1 of Cell-1, the parasitic C and R values of BL1 changes to 8.9fF and 1452 ohms whereas Node Q0 absorbed. And at other nodes, no change in the parasitic R and C values.

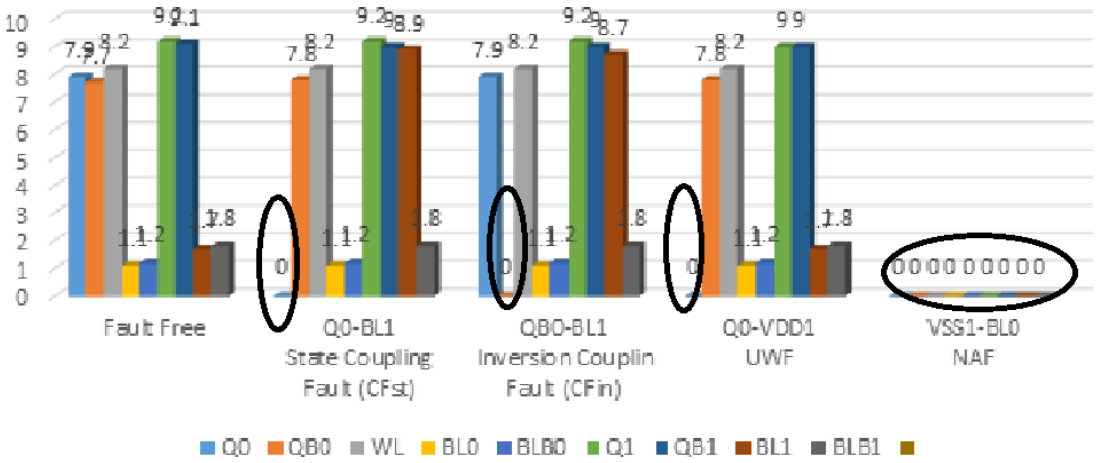


Fig 3.29a. Variation in Capacitance for different linked faults

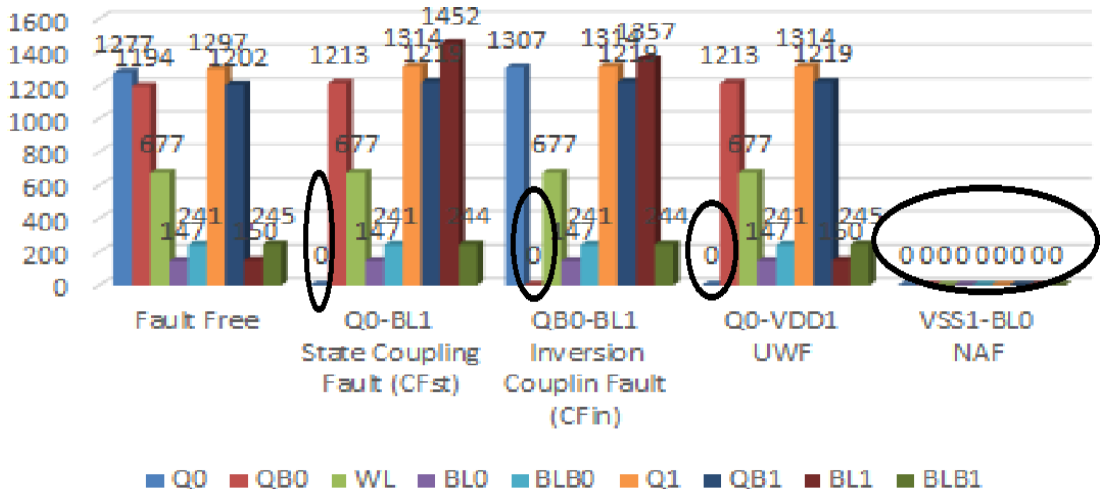


Fig 3.29b. Variation in Resistance for different linked faults

The graphical representation of fault detection based on parasitic capacitance variation using is shown in Fig.3.29b. Fault model NAF is modeled by short between VSS1 and BL0. As NAF does not allow the read and write operation, all nodes are absorbed. Hence for NAF the parasitic capacitance becomes zero.

Short between Q0-BL1 causes state coupling fault, this can detect by extracting R and C values at node Q0 .as shown in the figure the parasitic capacitance become zero at node Q0. Whereas at other nodes no change in the parasitic R and C values.

Similarly for the fault model inversion coupling fault corresponding to short defect modeled by short between QB0- BL1, in which the ‘parasitic capacitance and resistance’ variation is more affect at the node QB0.

For defect modeled by short between Q0-VDD1 which causes ‘Undefined Write Fault’, the ‘parasitic R,C; variation is observed only at node Q0., whereas these values remains same for other nodes.

Chapter 4

Machine Learning Based Parasitic Extraction Method

Very Large Scale Integration (VLSI) design has seen significant transformation in recent years as a result of the incorporation of machine learning (ML) techniques. Machine learning, a subset of artificial intelligence, has transformed many sectors, including VLSI design. Machine learning algorithms have proven to be essential in increasing the efficiency, accuracy, and productivity of VLSI design processes.

4.1 Need of Machine Learning in VLSI Design

The use of machine learning in VLSI design has numerous advantages. For starters, machine learning automates various labor-intensive procedures, lowering the time and effort necessary for design improvement. ML algorithms can rapidly evaluate massive volumes of data, helping designers to make more educated decisions and discover potential design problems.

Second, machine learning improves VLSI circuits' overall performance and power efficiency. ML algorithms may forecast and optimize numerous factors in the design process by evaluating and learning from historical data, resulting in speedier and more energy-efficient circuits. This not only enhances the end-user experience but also lowers overall power consumption, making VLSI systems more environmentally friendly.

Finally, machine learning makes it easier to experiment with new design ideas. Designers can build unique circuit topologies and explore atypical design spaces by employing ML algorithms. This brings up new creative possibilities and enables the development of hitherto undiscovered cutting-edge VLSI solutions.

4.2 Machine Learning Applications in VLSI

Machine learning has a wide range of applications in VLSI. One well-known application is the use of ML algorithms for VLSI layout optimization. Layout optimization has always been a time-consuming and error-prone process. Designers

can automate this process by using machine learning techniques, resulting in faster and more accurate layout designs. Machine learning algorithms can learn from existing layouts and develop optimal layouts that reduce power consumption while improving signal integrity.

Fault detection and diagnosis is another important application of machine learning in VLSI. VLSI circuits are prone to a variety of errors, which can result in system breakdowns and performance degradation. Large amounts of circuit data can be analyzed by ML algorithms to find anomalous patterns that signal the presence of problems. Machine learning can assist increase the reliability and robustness of VLSI circuits by effectively detecting and diagnosing errors.

Furthermore, machine learning can be used to forecast and optimize VLSI circuit performance. ML algorithms may learn from prior data and accurately forecast a circuit's performance under various operating situations. This enables designers to optimize circuit characteristics and efficiently attain the specified performance targets.

4.3 Improving VLSI Design with Machine Learning Algorithms

Machine learning techniques have shown to be beneficial in the overall design of VLSI circuits. The automated development of optimal designs is one method machine learning improves VLSI design. Algorithms can produce unique and optimal architectures that surpass traditional manual designs by training ML models on a huge dataset of current designs.

Another area where machine learning algorithms might help with VLSI design is power consumption optimization. ML models are capable of analyzing circuit parameters and identifying power-hungry components or inefficient designs. Machine learning allows designers to cut power consumption without sacrificing performance by providing insights into power optimization approaches.

Furthermore, machine learning methods can help identify potential design faults and circuit vulnerabilities. ML models can forecast possible faults or performance bottlenecks in VLSI circuits by evaluating past data and recognizing patterns. This proactive method enables designers to address these concerns early on, resulting in

fewer design iterations and improved overall circuit reliability.

4.4 Machine Learning Techniques for VLSI Layout Optimization

VLSI layout optimization is an important phase in the design process since it affects circuit performance and power consumption directly. Machine learning approaches provide unique layout optimization solutions, allowing designers to create more efficient and dependable layouts.

The employment of genetic algorithms is a popular machine-learning technique for VLSI layout improvement. To develop optimal solutions, genetic algorithms replicate the process of natural selection and evolution. Genetic algorithms can efficiently search the design space and create solutions that match the required objectives, such as lowering power consumption or optimizing performance, by storing the layout parameters into a chromosome-like representation.

Reinforcement learning is another machine learning technique used in VLSI layout optimization. Reinforcement learning agents learn by trial and error, with feedback from their surroundings. Reinforcement learning agents can explore multiple layout configurations and receive feedback on their performance in the context of VLSI layout optimization. Over time, the agents learn to develop layouts that perform better and adhere to certain design requirements.

Convolutional neural networks (CNNs) have also demonstrated promise in VLSI layout optimization. CNNs excel in spatial data analysis, making them an excellent alternative for analyzing and optimizing VLSI design. Designers can use the learned knowledge from training CNNs on current layout datasets to develop layouts that are better optimal for performance, power consumption, and area.

4.5 Machine Learning-Based Fault Detection and Diagnosis in VLSI Circuits

Fault detection and diagnosis are critical elements in assuring VLSI circuit reliability and resilience. Machine learning-based techniques have been demonstrated to be useful in automatically detecting and diagnosing errors, hence enhancing the overall fault tolerance of VLSI designs.

Anomaly detection is one method for detecting faults using machine learning. Algorithms for anomaly detection learn from normal circuit behavior and discover deviations that signal the existence of problems. ML algorithms can detect anomalies and indicate potential defects by assessing numerous circuit metrics such as voltage levels, current flows, and timing characteristics.

Fault diagnosis, on the other hand, entails determining the core cause of a problem. Machine learning algorithms can learn to link distinct fault patterns with their related causes by analyzing circuit activity during problematic scenarios. ML algorithms may accurately discover defects by using this learnt information, decreasing the time and effort necessary for manual diagnosis.

Furthermore, machine learning methods in VLSI circuits can provide proactive defect prevention. ML algorithms may forecast possible failure locations in a circuit and offer design changes to avoid these risks by evaluating past failure data and recognizing patterns. This proactive strategy contributes to the overall dependability and longevity of VLSI circuits.

4.6 Challenges and Limitations of Machine Learning in VLSI

Despite its promise possibilities, machine learning in VLSI design is fraught with difficulties and constraints. The availability of high-quality training datasets is a key barrier. For training, machine learning models rely substantially on vast and diverse datasets. However, due to private designs and confidentiality considerations, getting such datasets in the field of VLSI design might be difficult. The scarcity of training data can impair the performance and generalization capacities of machine learning systems.

The difficult interpretation of machine learning algorithms is another issue. VLSI designs necessitate a high level of transparency and explainability, as designers must comprehend the reasoning behind ML models' decisions. Many machine learning methods, for example deep neural networks, are fundamentally black-box models, making interpretation challenging. This lack of interpretability might be a major constraint in key VLSI design jobs.

Furthermore, the computational complexity of machine learning algorithms provides a barrier in VLSI design. VLSI circuits frequently have stringent timing constraints and must be processed in real time. However, many machine learning methods, particularly deep learning models, are computationally demanding and may not match the VLSI design's real-time requirements. To address this constraint, effective deployment and implementation of machine learning models in the VLSI domain are required.

In conclusion with machine learning potential to automate labor-intensive operations, boost performance, and improve fault tolerance, machine learning has become a vital tool in VLSI design. Machine learning has numerous applications in VLSI, spanning from layout optimization to defect identification and diagnosis. Despite the obstacles and restrictions, the future opportunities of machine learning in VLSI design are bright, with potential breakthroughs in deep learning, integration with upcoming technologies, and addressing edge computing and IoT demands. Machine learning is going to play an important part in defining the future of VLSI design as the discipline evolves.

4.7 Machine Learning Techniques in embedded Memory

The researchers presented artificial intelligence (AI) techniques in the expanding domain of chip design in the VLSI and automation sector. Their goal was to find a way to circumvent the challenges that arose during the various stages of development and design. At the beginning of the process, the AI procedures, such as knowledge-based and skilled systems, attempt to state the problem and then select the most appropriate result from a field filled with a variety of possible solutions. There has been a rapid and extraordinary development in the ever-growing VLSI technology as a result of the incorporation and involvement of the most recent design automation tools. This upgradation from the design of VLSI chips to the design of Ultra Large Scale Integrated circuit systems has also occurred as a result.

The VLSI industry is the only one capable of fabricating the various kinds of hardware that are necessary for artificial intelligence. One could say that the VLSI field and the field of artificial intelligence are intertwined with one another beginning with a wide

variety of applications such as remote controls, washing machines, cell phones, microwave ovens, air conditioners, car electronics, spaceships, aviation, weather forecast satellites, and defense, electronics have permeated every aspect of modern life. Every day, the race toward digitization has called for the development of new electronic systems that have low power consumption, higher battery backup, low cost, the fastest computational speed, and very short design times.

Because the size of the components continues to decrease on a daily basis, the research that is responsible for designing all of these electronic devices needs to be modernized at a faster rate. In the years to come, the VLSI industry will experience a significant upswing. It is necessary to introduce methodologies to reduce the complexity of the design in order to reduce the irregularities in the design while growing the chip in order to enhance the apparent growth in the nanometer range in the integrated circuit industry. This will allow for an increase in the apparent growth of the industry. The turnaround time of chip manufacturing needs to be shortened as quickly as possible. This is the single most important goal of the design process. Outdated methods, which were primarily manual and not automated, were used for those employed for such responsibilities; as a result, the processing took a greater amount of time, and the process consequently became very time-consuming and resource-intensive. The exclusive strategies of artificial intelligence (AI) offer several exhilarating methods for handling complex and data-concentrated tasks in the design and testing of VLSI, in comparison to the older methodologies. These tasks include the design and testing of VLSI. By embedding and incorporating the most recent techniques in the design of VLSI and manufacturing, it is possible to eliminate the complications that arise during the process as well as the delays that result from them. The procedures that are implemented make use of the automated learning algorithms of Artificial Intelligence and Machine Learning. This assists in reducing the amount of time and effort required for comprehending and processing the information.

The end result improves the production of integrated circuits while simultaneously reducing the amount of time needed for manufacturing turnaround. The technology that was used to design the system to overcome the overall design constraints will

have a significant impact on how much of an improvement there is in the turnaround time for chips in general. Electronic design automation is a tool that can be used to produce the best possible solution for the design constraints that have been set.

Recent developments in the application of machine learning approaches to design research challenges have generated a lot of interest [10, 11]. A model is trained or guided by the actual application of a process or phenomena, and then it is used to predict the same metric for new input data. The training set refers to the data used to develop the model initially. It should be evaluated using an entirely new set, known as the testing set, in order to determine the goodness of the developed model. If the actual set of inputs chosen is highly linked with the expected output, it is crucial to consider the fitness value of the training set.

We must have a solid and broad training set from real data obtained through operations in order to have a good model for variation estimates. To achieve this, the researcher have obtain huge data set for the short and open faults. For short faults totally the researcher got 21 defects at 7 nodes. Each node will have different resistance and capacitance values for different faults. Similarly the researcher have calculated the parasitic R, C value for 25 different open faults at 7 nodes. Table 3 and table 5 shows the obtained values. This will provide a large dataset for training and testing. One of the key features of our work is the use of actual layouts to extract parasitic R, C values, then the researcher impose the short/open defects then calculated the Parasitic R, C values, these values are used find the defects of the SRAM cell.

Machine learning is used to effectively learn from data and has been shown to address issues in a variety of fields. This thesis uses machine learning to verify digital circuit designs, generate test patterns, and discover faults. This research effort was carried out using the supervised machine learning technique. Two forms of supervised machine learning approaches are linear regression and classification. The behavioral design verification, defect detection, and test pattern creation were all verified using both models. To anticipate the test pattern and the total number of faults, the linear regression model was utilized. The classification model was used to validate a circuit's behavioral logic.

The history of machine learning is provided in this chapter. It comprises many machine learning methodologies as well as the gradient descent algorithm for ML model convergence. Below are several definitions of machine learning. We currently live in a highly technical society as a result of the emergence of VLSI and embedded systems. To work correctly, the majority of electronic gadgets that we use today need computing. Millions of transistors may now be manufactured on a single chip thanks to recent advancements in VLSI technology, enabling for more efficient computing. Higher power consumption on a chip is a consequence of increasing the number of transistors and operating speeds on a chip, which has become a severe problem in the submicron technology sector. In VLSI circuit design, getting an accurate estimate of power early on is critical since it has a significant influence on circuit dependability and reliability. In this scenario, obtaining an average power estimate prior to chip production is critical because it enables designers and engineers to calculate a power budget and take the required steps to decrease power consumption at higher levels of design abstraction. The goal of this study is to provide a less difficult and low-cost power estimating method that may be used instead of known approaches like benchmark circuit modeling, which are dependent on the assumption of pre-determined empirical equations and hence less accurate.

4.8 Machine Learning Design Methodology

Machine learning is a branch of artificial intelligence that allows systems to learn from large amounts of data and address certain issues. It makes use of computer algorithms whose effectiveness is automatically improved through practice.

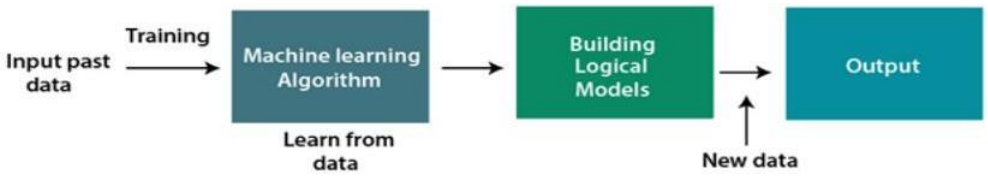


Fig 4.1. Working model of Machine Learning algorithm

There are primarily three types of machine learning: Supervised, Unsupervised, and Reinforcement Learning.

4.8.1 Supervised Learning: In supervised learning, machine learning models are trained using labeled data. The outcome in labeled data is already known. The model only needs to map the inputs to the corresponding outputs. Algorithms for supervised learning are frequently employed to solve classification and regression issues.

Linear Regression, Logistic Regression, SVM algorithm, KNN algorithm, Decision Tree, Random Forest are supervised learning algorithms

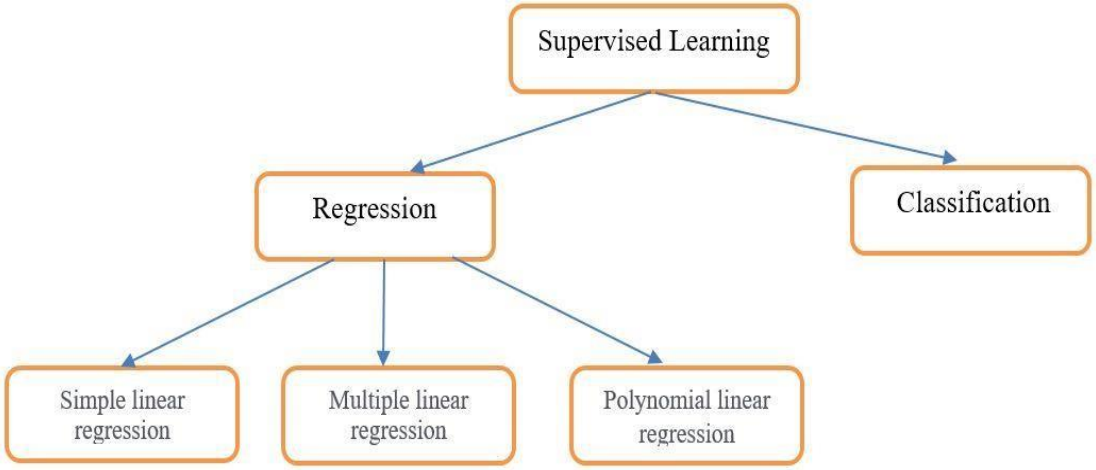


Fig 4.2. Types of Supervised Learning

4.8.2 Unsupervised Learning: Machines are trained with unlabeled data using a technique called unsupervised learning. No fixed output variable exists for unlabeled data. The model takes in the information from the data, looks for patterns and features, and then outputs the results. For the purpose of resolving clustering and association issues, unsupervised learning is employed.

4.8.3 Reinforcement Learning: Reinforcement Learning enables a machine to respond appropriately and maximize its benefits in a certain circumstance. To generate actions and rewards, it makes use of an agent and an environment. The agent has a beginning state and a conclusion state. However, there could be numerous routes leading to the goal, much like a maze. There is no fixed target variable in this learning

method.

In our proposed method, the researcher have used multiple linear regression to predict the parasitic R, C values. The regression method explained in the following section.

Simple Linear Regression:

The relationship between independent and dependent variables can be predicted using a statistical model called linear regression by looking at two aspects:

1. Specifically, which variables are capable of accurately predicting the outcome variable?
2. In terms of creating predictions with the highest degree of accuracy, how significant is the regression line?
3. An independent variable's value is unaffected by the effects of other variables. It is frequently indicated with a "x."

The dependent variable is affected by an independent variable. When the values of the independent variables change, the dependent variable's value also changes. It is frequently indicated by a "y".

The linear regression represented by the equation of $y = m*x + c$

Where x → independent variable, y → dependent variable, m → slope

Multiple Linear Regression

The multiple linear regression, represented by the equation of $y = m_1x_1 + m_2x_2 + m_3x_3 + ... + c$

Where x_1, x_2 and $x_{3...}$ are the independent variables. m_1, m_2, $m_3$indicates the slopes.

4.9 Determination of Parasitic R, C values by using Multiple Linear Regression:

Table 21. Shows the extracted R, C values for the different technologies from the layout diagram of the 6T-SRAM Cell at each node as shown in the fig 1. In the table shown the researcher have used multiple linear regression to determine the R, C values. In this process the researcher have used technology and length as the independent variables and Resistance and Capacitance are the dependent variables.

Table 4.1Extracted R, C values for fault free single 6T-SRAM cell using different technologies

Node	Technology(nm)	L(um)	R(ohms)	C(fF)
Q	120	69.2	1336	7.1
QB	120	78.3	1415	7.5
WL	120	31.3	371	4
BL	120	20.7	146	0.973
BLB	120	28.3	243	1.2
VDD	120	12.6	2	0.604
VSS	120	12.6	2	0.604
Q	90	64.7	1013	6.8
QB	90	57.8	949	6.5
WL	90	21.1	337	2.9
BL	90	18.1	99	1.2
BLB	90	10.2	188	0.753
VDD	90	9.4	6	0.537
VSS	90	9.4	6	0.537
Q	45	29.9	1518	3.6
QB	45	25.8	1128	3.3
WL	45	9.6	415	1.4
BL	45	8.1	152	0.816
BLB	45	4.6	247	0.484
VDD	45	4.2	8	0.404
VSS	45	4.2	8	0.404
Q	32	19.4	818	2
QB	32	17	682	1.8
WL	32	7.3	335	0.692
BL	32	6.3	75	0.701
BLB	32	3.9	67	0.459
VDD	32	2.9	13	0.314
VSS	32	2.9	13	0.314
Q	7	3.3	7417	0.642
QB	7	3.5	7077	0.681
WL	7	1.7	3553	0.34
BL	7	1.3	951	0.195
BLB	7	1.7	1371	0.256
VDD	7	0.65	23	0.081
VSS	7	0.65	23	0.081

The main steps involved in the multiple linear regression to determine the R and C values are as follows:

- Importing the libraries
- Load the data set and extract independent and dependent variable
- Data Visualization
- Encoding the Data
- Splitting the data into train and test set
- Fitting the Multiple Linear Regression to training set
- Predicting the test results

Simulation Results:

1. Importing the libraries: Imported the Pandas and Numpy libraries for the data processing and perform the numerical operations respectively.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

2. Load the data set and extract independent and dependent variable

```
dataset = pd.read_csv('C:\\Users\\User\\Desktop\\RCValues.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
y
```

```
array([7.1 , 7.5 , 4.  , 0.97, 1.2 , 0.6 , 0.6 , 6.8 , 6.5 , 2.9 , 1.2 ,
       0.75, 0.54, 0.54, 3.6 , 3.3 , 1.4 , 0.82, 0.48, 0.4 , 0.4 , 2.  ,
       1.8 , 0.69, 0.7 , 0.46, 0.31, 0.31, 0.64, 0.68, 0.34, 0.2 , 0.26,
       0.08, 0.08])
```

pd.read_csv is used to load the data. Dataset. iloc is used to select the particular row and column to determine the dependent and independent variables. In the given dataset resistance column taken as the dependent variable and technology and length as independent variables.

3. Data Visualization and Encoding the data

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers =[('encoder', OneHotEncoder(), [0])], remainder = 'passthrough')
X = np.array(ct.fit_transform(X))
X
```

```
array([[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 120, 69.2, 1336],
       [0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 120, 78.3, 1415],
       [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 120, 31.3, 371],
       [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 120, 20.7, 146],
       [0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 120, 28.3, 243],
       [0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 120, 12.6, 2],
       [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 120, 12.6, 2],
       [0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 90, 64.7, 1013],
       [0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 90, 57.8, 949],
       [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 90, 21.1, 337],
       [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 90, 18.1, 99],
       [0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 90, 10.2, 188],
       [0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 90, 9.4, 6],
       [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 90, 9.4, 6],
       [0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 45, 29.9, 1518],
       [0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 45, 25.8, 1128],
       [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 45, 9.6, 415],
       [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 45, 8.1, 152],
       [0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 45, 4.6, 247],
       [0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 45, 4.2, 8],
       [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 45, 4.2, 8],
       [0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 32, 19.4, 818],
       [0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 32, 17.0, 682],
       [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 32, 7.3, 335],
       [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 32, 6.3, 75],
       [0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 32, 3.9, 67],
       [0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 32, 2.9, 13],
       [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 32, 2.9, 13],
       [0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 7, 3.3, 7417],
       [0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 7, 3.5, 7077],
       [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 7, 1.7, 3553],
       [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 7, 1.3, 951],
       [0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 7, 1.7, 1371],
       [0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 7, 0.65, 23],
       [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 7, 0.65, 23]], dtype=object)
```

4. Splitting the data into train and test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 20, random_state =
```

5. Fitting the Multiple Linear Regression to training set

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

6. Predicting the test results

```
y_pred = regressor.predict(X_test)
np.set_printoptions(precision = 2)
print(np.concatenate((y_pred.reshape(len(y_pred),1),y_test.reshape(len(y_test),1)),1))
y_pred
```

```
[[ 0.78  0.68]
 [ 0.44  0.4 ]
 [ 1.45  1.4 ]
 [ 1.35  0.64]
 [ 1.29  1.8 ]
 [ 2.22  3.3 ]
 [ 1.19  1.2 ]
 [ 3.14  4.  ]
 [ 0.03  0.75]
 [ 0.44  0.31]
 [-0.08  0.46]
 [ 0.45  0.08]
 [ 0.09  0.26]
 [ 0.29  0.31]
 [ 1.33  0.34]
 [ 5.4   6.5 ]
 [ 0.56  0.54]
 [ 0.46  0.6 ]
 [ 0.53  0.82]
 [ 3.3   3.6 ]]
```

```
array([ 0.78,  0.44,  1.45,  1.35,  1.29,  2.22,  1.19,  3.14,  0.03,
        0.44, -0.08,  0.45,  0.09,  0.29,  1.33,  5.4 ,  0.56,  0.46,
        0.53,  3.3 ])
```

```
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

```
0.8862053149724555
```

Thus our proposed model gives the 88.62% accuracy to determine the parasitic C values for fault free SRAM Cell. The extracted Parasitic R, C values used to find the defects and location of the faulty SRAM cell. Table 1 shows fault model dictionary for all open faults. Table 4 shows fault model dictionary for all short faults.

After estimation of the parasitic R, C Values, These values have used Decision Tree algorithm to find the fault and its location, A decision tree is a tree-based supervised learning technique used to forecast a target variable's result. With the support of regression and classification algorithms, supervised learning employs labeled data information with known output variables to create predictions. Using different data features, it learns from basic decision-making guidelines. Python decision trees are widely used to calculate probabilities because they may be utilized to handle classification and regression issues.

Important Terms Used in Decision Trees:

1. **Entropy:** The amount of uncertainty or randomness in a set of data is measured by entropy. How a decision tree divides the data depends on entropy. The following formula is used to calculate the uncertainty.

$$\sum_{i=1}^{k} P(value_i).log_2(P(value_i)$$

2. **Information Gain:** After the data set is divided, the information gain calculates the reduction in entropy.

IG(Y, X) = Entropy (Y) - Entropy (Y | X) formula is used to calculate the information gain

3. **Gini Index:** To select the appropriate variable for splitting nodes, the Gini Index is used. It assesses the frequency of inaccurate identification of a randomly selected variable.

4. **Root Node:** The top node of a decision tree is always the root node. It can be further split into various sets and represents the total population or data sample.

5. **Decision Node:** Decision nodes are sub nodes that can be divided into other sub nodes and include two or more branches.

6. **Leaf Node:** A leaf carries the final results. These nodes, are also known as terminal nodes, and these nodes further cannot be split any further

The decision tree algorithm in machine learning used to predict if the memory cell is faulty cell or fault free cell. In order to make the prediction, the data set includes a variety of information, such as the capacitance and resistance values for defective and fault-free SRAM cells at each node, as well as fault information.. Table shows the extracted capacitance and resistance values for the open fault detection at the nodes QB, Q, WL, BL, BLB and VDD.

Table 4.2. Extracted R and C values for faulty and fault free SRAM at different nodes

C in fF	R in KΩ	Faulty or Not	C in fF	R in KΩ	Faulty or Not	C in fF	R in KΩ	Faulty or Not
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.26	1.36	Fault at WL
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.8	1.36	Fault at WL
2.9	18.96	Fault Free	2.4	12.26	Fault at Q	0.88	2.66	Fault Free
2.9	18.96	Fault Free	2.7	16.74	Fault at Q	0.88	2.66	Fault Free
2.9	18.96	Fault Free	1.2	3.62	Fault at Q	0.88	2.66	Fault Free
2.9	18.96	Fault Free	2.8	16.13	Fault at Q	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3	18.94	Fault at Q	0.88	2.66	Fault Free
2.9	18.96	Fault Free	2.6	14.91	Fault at Q	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.1	11.06	Fault at QB	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.4	15.51	Fault at QB	3.3	20.18	Fault Free	0.88	2.66	Fault Free
0.52	2.31	Fault at QB	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.5	14.94	Fault at QB	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.7	17.74	Fault at QB	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.7	17.74	Fault at QB	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.3	13.72	Fault at QB	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	2.6	14.91	Fault at Q	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0.88	2.66	Fault Free
2.9	18.96	Fault Free	3.3	20.18	Fault Free	0	0	Fault at WL
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
0	0	Fault at BL	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.1	8.65	Fault at VDD
1	2.6	Fault Free	0.61	3.29	Fault Free	2.1	8.65	Fault at VDD

1	2.6	Fault Free	0.61	3.29	Fault Free	1.9	5.27	Fault at VDD
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.29	Fault Free	2.4	12.03	Fault Free
1	2.6	Fault Free	0	0	Fault at BLB	2.4	12.03	Fault Free
1	2.6	Fault Free	0.61	3.3	Fault Free	2.4	12.03	Fault Free

4.10 Building a Decision Tree for fault detection in SRAM Cell

1. Import the libraries for Decision Tree.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
import matplotlib.pyplot as plt
```

2. Load the data using Pandas

```
rc_data = pd.read_csv('C:\\Users\\User\\Desktop\\Open Faults RC Values.csv', sep = ",", header =
rc_data.head()
```

	C in fF	R in Kohms	Faulty or Not
0	2.9	18.96	Fault Free
1	2.9	18.96	Fault Free
2	2.9	18.96	Fault Free
3	2.9	18.96	Fault Free
4	2.9	18.96	Fault Free

3. Slicing method separate dependent and independent variables.

```
X = rc_data.values[:,0:2]
y = rc_data.values[:,2]
print(y)
```

4. Using the decision tree classifier split the train and test data

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.4,random_state=10)
clf_entropy = DecisionTreeClassifier(criterion = "entropy",random_state=10,max_depth =3,min_samples_leaf =
clf_entropy.fit(X_train,y_train)
```

5. Predict the test data set values.

```
y_pred = clf_entropy.predict(X_test)
y_pred
```

```
array(['Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault at WL', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free', 'Fault Free', 'Fault Free', 'Fault Free',
       'Fault Free'], dtype=object)
```

6. Calculate the accuracy of the model.

```
print("accurcy is "), accuracy_score(y_test,y_pred)*100
```

```
accurcy is

(None, 91.78082191780823)
```

Therefore our prediction model shows that there is an excellent accuracy score of 91.78 percent to separate faulty memory cells and also locate the position of the defect irrespective of the technology variation.

Chapter-5

Conclusion Future Scope

Semiconductor memory testing increasing demand due to advances in mobile phones, smart phones, washing machines, digital cameras and refrigerators etc.. To save the data the digital system requires more memory, hence the SoC becomes memory dominant. As device downsizing, the number of transistor count increases, the occurrence of faults increases and testing of memory become essential. Parasitic Extraction method for embedded memory is proposed for open and short faults. Using the three technologies 45nm, 32 nm and 7nm the fault models are developed for open and short faults. The fault models developed for open defects are 25 for each technology. Therefore the overall fault models implemented for open faults are 75. Similarly the fault models developed for short defects are 21 for each technology. Therefore the overall fault models implemented for open faults are 63.

Along with existing faults like Stuck at Faults, Transition Faults, No Access Faults, Undefined Read Fault, Undefined Write Faults, Initialization Order Fault, Unstabilized Read Fault, Unstabilized Write Fault, Write Before Access Fault, The researcher have observed a new fault for the short defects for 45nm technology named it as Undefined Short Fault.

Fault Model Dictionary is developed using Parasitic Extraction method at the chosen technology levels 45nm, 32nm and 7nm. Few fault models are consistent with their fault behaviour in three technologies. As an example for short defect, for the fault model SD2 (short between WL and BL) using 45nm technology the cell behaves Stuck At one fault and behaves transition Faults for 32nm technology and Write Before Access Fault for 7nm technology. Similarly for fault model SD3 (short between WL-BLB) gives the new undefined short fault, same fault model behaves unstabilized read fault for 32 nm technology, and observes write before access fault for 7nm technology. Similarly for fault model SD10 (short between Q and VSS), SD12(short between QB-VSS), SD14(short between QB and BLB, cell behaves differently for different technologies.

The Parasitic Extraction method also used to detect the linked faults in the two cell SRAM. Faults like State Coupling Fault, Inversion Coupling Fault, Undefined Write Fault, Ustabilized Write Fault observed. Fault Model Q0-QB10 (short between Q0 and QB10), Q0-BLB10, Q0-VDD1, VDD0-Q1, QB0-BL1, QB0-QB10, QB0-VDD1, and VSS0-Q1gives the fault masking. Fault Masking means cell performs the correct operation, even though node have the defect. Thus the proposed parasitic extraction method gives the 100 percent fault coverage.

Method of Parasitic extraction is discussed in terms node parasitics as well as instance parasitics based on the test environment chosen. Using 75 models for open defects, 63 fault models for short defects, extracted their corresponding defect induced layouts, extracted node as well as instance parasitics. The extracted parasitics R and C are compared with the parasitics of prototyped fault free SRAM layout for the observation of fault affected node. Table represent the complete fault model dictionary using 45nm, 32 and 7 nm technology for single 6T SRAM cell R, C values respectively. Tables is an observation for unique faults (whose parasitic R, C are different from Fault Free R, C) list for 45nm, 32nm and 7 nm respectively by considering input output nodes alone.

In spite of the fact that test time and fault coverage have dependably been real concerns, the industry fundamentally depends on developed algorithms for testing memory chips. The development of system chips brought forward new issues for analysts. Both the quantity of embedded memory chips and area occupied by memories are quickly increasing on embedded devices. The yield of on-chip memories subsequently decides device yield. So, memory testing is rapidly turning into a basic issue, the extent that assembling yield and time-to- volume of embedded device are concerned. The increase in memory volume in embedded devices make the memory testing complex. All the fundamental fault models need to be covered and some other issues arise due to higher volumes, scaling of transistors and large number of memory cores need to be addressed. Issues like high power consumption, complexity, area occupied by the test circuitry need to be addressed.

Machine Learning allows Machine automatically learn from past data without programming explicitly. The goal of ML is to allow machines to learn from data so that they can give accurate output Machine learning is mainly concerned about

accuracy and patterns supervised learning algorithms used to predict the parasitic R, C values and to detect the faults.

The huge data set required, in order to train the machine. The researcher have prepared the dataset for 120nm, 90nm, 45nm, 32nm and 7nm technologies. Table shows the extracted values of parasitic R, C and length from the layout of the each technology using microwind. Multiple Linear Regression used to predict the Parasitic R, C. In this Length and technology taken as independent variable and Parasitic R, C are dependent variable. The results shows the 88.62 percentage of accuracy to predict the R and C values. Decision Tree algorithm used find the fault along with location. Table shows the data set of Extracted R, C values at each node for the different technologies 45nm, 32nm and 7 nm. The results shows the 91.78 percent to separate faulty memory cells and also locate the position of the defect irrespective of the technology variation.

Table 4.3. Fault Coverage Comparison of different methods

S.No	**Fault Type**	**March C-**	**March SS**	**March LR**	**wavelet-based transient supply current testing**	**March ML3S algorithm**	**Proposed Parasitic Extraction Method**
1	Stuck-At (SAF)	√	√	√	√	√	√
2	Transition (TF)	√	√	√	√	√	√
3	Read Destructive (RDF)	√	√	√	√	√	√
4	Incorrect Read (IRF)	√	√	√	√	√	√
5	Deceptive Read Destructive (DRDF)	√			√	√	√
6	Undefined Read Fault	√		√	√	√	√
7	Undefined Write Fault	√		√	√	√	√
8	Unstabilized Read fault					√	√
9	Unstabilized write fault					√	√
10	Initialization Order fault						√
11	Write Before Access Fault						√
12	Read Destructive Retention Fault						√
13	State Coupling Fault (CFst)	√				√	√
14	Inversion Coupling Fault (CF_{in})	√				√	√

15	Inaccessible State Coupling Fault (CFist)						√
16	Transition Coupling (CFtr)	√	√	√		√	√
17	Deceptive Read Destructive Coupling (CFdrd)					√	√
18	Incorrect Read coupling fault (CFir)		√			√	√
18	Fault Masking						√

Table 4.4. Comparison between different Machine Learning algorithms

Machine Learning Applications for the Proposed Parasitic R and C Extraction Method.	**Machine Learning Applications in Physical Design Recent Results and Directions Statistical and Machine Learning-Based CAD**	**MLParest: Machine Learning based Parasitic Estimation for Custom Circuit Design**	**Machine LearningBased Variation Modeling and Optimization for 3D ICs.**
1.Multiple Linear Regression Machine learning algorithms used to estimate the Parasitic R and C Values for different technologies(45nm, 32nm, 7nm) for 6T SRAM Cell, with the accuracy of 88.62% 2.Decision Tree is used to find the faults in the 6T SRAM cell, by using extracted R and C values with the accuracy of 91.7%	Discussed different Machine Learning algorithms for IC Design applications Example applications include (1).Removing unnecessary design and modeling margins through correlation mechanisms. (2) Achieving faster design convergence through predictors of downstream flow outcomes that comprehend both tools and design instances.	1.Parasitic estimation (MLParest) method for pre-layout custom circuit design is presented. 2.It reduces the error between pre- layout and post-layout circuit simulation from 37% to 8% on average for different measurements across a variety of analog circuits. 3.MLParest can thus greatly reduce the number of iterations between pre-layout and post-layout design phases.	1.Developed a new machine learning based model and methodology for an accurate variation estimation of logic paths in 3D designs. 2.Uses key parameters extracted from existing GDSII 3D IC design and sign off simulation database.

Future Enhancements

The most recent popular phrases, such as artificial intelligence (AI), machine learning (ML), and deep learning (DL), do not anymore belong to the world of information technology (IT) or software development. Some of the most popular engineering subfields, such as Electrical and Electronics Engineering and Electronics and Communication Engineering, have a close link. Any substantial improvement in software technology necessitates equivalent advances in hardware to effectively support it.

To have a basic understanding of the phrase, "machine learning" corresponds to an artificial intelligence system capable of self-learning based on an algorithm. Machine learning and deep learning are examples of how systems get more intelligent over time without the assistance of people. Deep learning is the application of machine learning to big data sets, involving additional layers for data processing.

These terms are intertwined since the majority of artificial intelligence work involves machine learning. This is related to the fact that intelligent behavior necessitates a considerable amount of knowledge in order to absorb new approaches. New machine learning algorithms are being implemented in diverse ways, and hardware systems will swiftly adapt to accommodate these changes.

There are also new architectures available, some of which use more recent data processing technologies. Machine learning will necessitate improved custom hardware to stay up with the quickly evolving designs as the times change. Machine learning has a very broad range of applications, as does its scope. A big number of inventors and developers are already working on various sorts of upgrading on the market. Algorithms, model training, rules, and other similar things are the foundations of the systems and technologies, areas in which software and computer engineering play a vital part. However, it is important to recognize that the program requires very high-end hardware, has a substantial compute capacity, and consumes less power, and is able to carry out complex mathematical functions in a matter of fractions of microseconds.

References

1. Semiconductor Industry Association (SIA), "International Technology Road map for semiconductors (ITRS)", 2003.

2. J. F. Li, K. L. Cheng, C. T. Huang and C. W. Wu, "March-based RAM diagnosis algorithms for stuck-at and coupling faults", Proceedings of IEEE International Test Conference, pp. 758–767, 2001.

3. M. Klaus, A. J. Van de Goor, "Test for resistive and capacitive defects in address decoders", Proceedings of IEEE Asian Test Symposium, pp. 31–36, 2001.

4. A. Ney , A. Bosio , L. Dilillo , P. Girard , S. Pravossoudovitch , A. Virazel & M. Bastian, "A History-Based Diagnosis Technique for Static and Dynamic Faults in SRAMs", Proceedings of IEEE International Test Conference, pp. 81-83, 2008.

5. L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel and M.H. Hassan, "Resistive-open defect injection in SRAM core-cell: analysis and comparison between 0.13µm and 90nm technologies", Proceedings of Association for Computing Machinery (ACM) IEEE Design Automation Conference, pp. 857– 862, 2005.

6. Y.Zhao, L.Grenier, A.Majumdar, "Power Characterization of Embedded SRAMs for Power Binning", 30th IEEE VLSI Test Symposium (VTS), 2012.

7. S.K. Lu, Y.C. Hsiao, C.H. Liu, and C.L.Yang "Low Power Built In Self-Test Techniques for Embedded SRAMs", Hindawi Publishing Corporation VLSI Design Volume 2007.

8. L.B. Zordan, A. Bosio, L. Dilillo, P. Girard, A. Todri, A. Virazel & N.Badereddine, "On the Reuse of Read and Write Assist Circuits to Improve Test Efficiency in Low Power SRAMs ", IEEE International Test Conference (ITC), 2013

9. S.Panth, S.K. Lim, "Transition Delay Fault Testing of 3D ICs with IR-Drop Study", 30th IEEE VLSI Test Symposium (VTS), 2012.

10. S.K.Ojha, P.R.Vaya, "A Novel Architecture of SRAM for Low power Application", International Journal of Advanced Electrical and Electronics Engineering, (IJAEEE), Volume-2, Issue-4, 2013.

11. M. Bastian, V. Gouin, P. Girard, C. Landrault, A. Ney, S. Pravossoudovitch, A. Virazel, "Influence of Threshold Voltage Deviations on 90nm SRAM CoreCell Behaviour", 16th IEEE Asian Test Symposium (ATS), 2007.

12. A.Calimera, A. Macii, E.Macii, M.Poncino, “Design Techniques and Architectures for Low Leakage SRAMs”, IEEE transactions on Circuits and Systems, Vol. 59, NO. 9, September 2012.

13. Y.Balasubrahamanyam, G.Leenendra Chowdary, T.J.V.S.Subrahmanyam, “A Novel Low Power pattern Generation Technique for Concurrent Bist Architecture ”, International Journal of Computer Technology & Applications, Vol 3 (2), 561-565, 2014.

14. M. Alioto, “Ultra-low power VLSI circuit design demystified and explained: A tutorial,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 59, no. 1, pp. 3–29, 2012.

15. D. Xiang, Y. Zhang, “Cost-effective power-aware core testing in NoCs based on a new unicast-based multicast scheme,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, no. 1, pp. 135–147, 2011

16. M. R. Rani, G. R. Kumar, G. P. Kumar, “Implementation of March Algorithm Based MBIST Architecture for SRAM,” International Journal of Advanced Research in Computer Engineering & Technology, vol. 1, no. 3, 2012

17. M.Parvathi, N. Vasantha, and K. Satya Prasad “Fault Model Analysis By Parasitic Extraction Method for Embedded SRAM”. International Journal of Research in Engineering and Technology, Dec-2013.

18. M.Parvathi, N. Vasantha, K. Satya Prasad “New Fault Model Analysis For Embedded Sram Cell For Deep Submicron Technologies Using Parasitic Extraction Method” International Conference on VLSI Systems, Architecture, Technology and Applications-2015.

19. M. Parvathi, K. Satya Prasad, N. Vasantha “Testing of Embedded SRAMs Using Parasitic Extraction Method” 9th International Conference on Robotic, Vision, Signal Processing and Power Applications, Lecture Notes in Electrical Engineering 398, 2016.

20. K. Sangwon, “Microcode-Based Memory BIST Implementing Modified March Algorithms,” Journal of the Korean Physical Society, vol. 40, no. 4, p. 749, 2013.

21. V.A.Vardanian, Y. Zorian, “A March-based fault location algorithm for static random access memories,” Records of the IEEE International Workshop on Memory Technology, Design and Testing, vol. 2002–January, pp. 62–67, 2002.

22. P.Caşcaval, D.Caşcaval, “March SR3C: A Test for a reduced model of all static simple three-cell coupling faults in random access memories,” Microelectronics Journal, vol. 41, no. 4, pp. 212–218, 2010

23. R.Wang, K.Chakrabarty, “A Programmable Method for Low Power Scan Shift in SoC Dies,” in Testing of Interposer-Based 2.5D Integrated Circuits, Springer, pp. 163–178, 2017.

24. M. Bushnell, V. Agrawal “Essentials of electronic testing for digital, memory, and mixed-signal VLSI circuits”, Springer Science & Business Media, vol. 17, no. 4. 2009

25. F. Saqib, J. Plusquellic, “VLSI test and hardware security background for hardware obfuscation,” in Hardware Protection 143 through Obfuscation, Springer, pp. 33–68, 2017.

26. L. Ali, R. Sidek, I. Aris, B. S. Suparjo, M. A. M. Ali, “Challenges and directions for testing IC,” Integration, the VLSI Journal, vol. 37, no. 1, pp. 17–28, 2004.

27. Benso, S.D.Carlo, G.D. Natale, P.Prinetto, “An optimal algorithm for the automatic generation of March tests,” in Proceedings - Design, Automation and Test in Europe, pp. 938–943, 2002.

28. G. R. Kumar, K. Babulu, “A Novel Architecture for Scan Cell in Low Power Test Circuitry,” Procedia Materials Science, vol. 10, pp. 403–408, 2015.

29. M. B. Josephs, “An analysis of determinacy using a trace-theoretic model of asynchronous circuits,” in Proceedings - International Symposium on Asynchronous Circuits and Systems, pp. 121–128, 2003.

30. Hamdioui, A.J. Goor, M. Rodgers, “ March SS – A test for all static simple SRAM faults”, Proceedings of the IEEE International Workshop on Memory Technology, Journal of Design and Testing (MTDT 2002), 95-100

31. S. Khursheed, “Test and Diagnosis of Resistive Bridges in Multi-Vdd Designs”, Ph.D. thesis, Departement of ECE, University of Southampton. 2010.

32. S. Velaga, “Fault Modelling Analysis for Multiple Voltage Power Supplies in Low Power Design”, University of Cincinnati, 2013.

33. L.Dilillo, P.Girard, S.Pravossoudovitch, A.Virazel, Borri, S.H. Hassan, “Resistive-Open Defects in Embedded-SRAM Core Cells: Analysis and March Test

Solution" IEEE, 13th Asian Test Symposium, 266–271. 2004.

34. M.T.Martins, G. Medeiros, T.Copetti, F.Vargas, L.B. Poehls (2016), "Analyzing NBTI Impact on SRAMs with Resistive-Open Defects" IEEE, 17th Latin-American Test Symposium, 87–92. 2016.

35. D.Arumí, R.-Montañés, J.Figueras "Test Escapes of Stuck-Open Faults Caused by Parasitic Capacitances and Leakage Currents", IEEE Transactions on Very Large Scale Integration Systems, 24(5), 1739–1748-2015.

36. Li Z., Lu X., Qiu W., Shi W, Walker D. "A Circuit Level Fault Model for Resistive Opens and Bridges". IEEE, In Proceedings. 21st VLSI Test Symposium, 379–384. 2003.

37. Azevedo J., Virazel A., Bosio A., Dilillo L., Girard P., T.Sanial A., Alvarez-Hérault J, Mackay K., "A Complete Resistive-Open Defect Analysis for Thermally Assisted Switching MRAMs", IEEE Transactions on Very Large Scale Integration Systems, 22(11), 2326–2335-2014.

38. Yu C., Liu G., Lai L., Diagnosis of Resistive-Open Defects Using IDDT in Digital CMOS Circuits, WSEAS Transactions on Circuits and Systems, 13, 296–300. 2014.

39. Renovell M., Comte M., Polian I., Engelke P, Becker B, "A Specific ATPG Technique for Resistive Open with Sequence Recursive Dependency". IEEE, 15th Asian Test Symposium, 273–278. 2006.

40. A.Czutro, N.Houarche,P.Engelke, I.Polian, M.Comte, M.Renovell, B.Becker," A Simulator of Small-Delay Faults Caused by Resistive - Open Defects. IEEE, 13th European Test Symposium, 113–118. 2008.

41. K.Yang, K.Cheng, L.Wang, "Tran Gen: A SAT- Based ATPG for Path-Oriented Transition Faults". In ASP-DAC 2004: IEEE Asia and South Pacific Design Automation Conference, 92–97. 2004.

42. M.B.Tahoori "Testing for Resistive Open Defects in FPGAs". IEEE International Conference on Field-Programmable Technology, Proceedings, 332–335. 2002.

43. N. Z. Haron, S.Hamdioui. "DFT Schemes for Resistive Open Defects in RRAMs" IEEE, Design, Automation & Test in Europe Conference & Exhibition, 799–804-2012.

44. R.R Montañés, J.P.D. Gyvez, P. Volf, "Resistance Characterization for Weak

Open Defects", IEEE Design & Test of Computers, 19(5), 18–26-2002.

45. M. A. Krishnan, C. G. Theepa, "Diagnosis of Resistive open Fault using Scan Based Techniques" International Journal of Engineering Research & Technology, 3(16), 1–5.2015.

46. L. Maharana, T.Sarkar, S.Pradhan, "Look up Table Based Low Power Analog Circuit Testing", International Journal of Engineering, 29(9), 1247–1256-2016

47. M. Chhillar, G.Yadav, N.K. Shukla, "Average and Static Power Analysis of a 6T and 7T SRAM Bit-Cell at 180nm, 90nm, and 45nm CMOS Technology for a High Speed SRAMs". In Proceedings in International Conference on Advances in Electrical & Electronics, 75–79, 2013.

48. R. Kumar, S. Choudhary, B. Prasad, "Simulation of 6T SRAM at 90nm and 180nm Technology and Study the Effect of Scaling on Read and Write Operation", Advanced Research in Electrical and Electronic Engineering, 1(4), 42–45-2014.

49. G. Shivaprakash. D.Suresh, "Design of Low Power 6T- SRAM Cell and Analysis for High Speed Application", Indian Journal of Science and Technology, 9(46), 1–10-2016.

50. S.Munaf, A.Jayanthi, "Review on Power Dissipation Analysis of Conventional SRAM Cell Architecture", International Journal of Advanced Research in Computer Engineering & Technology, 6(11)-2017.

51. N. Deora, P.Shrivastava, "Design Low Power 6T SRAM cell using Double Gate FINFET Technique", International Journal of Research and Analytical Reviews, 5(4), 66–75-2018.

52. E. R. Jennifer, A. S. Alima, Bharathisankari, "A Study on 6T SRAM and 7T SRAM Cells", International Journal of Research in Engineering and Technology, 7(2), 67–75-2018.

53. Y. S. Parihar, G.Jangid, "Design of a Low Power SRAM Cell by Tanner Tool 45nm", International Journal on Recent and Innovation Trends in Computing and Communication, 3(12), 6631–6636-2015.

54. M.Parvathi, N. Vasantha, K. Satya Prasad, " New Fault Model analysis for Embedded SRAM cell for Deep Submicron Technologies Using Parasitic Extraction Method", International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA), 2015.

55. K.Khare, N. Khare, V.K. Kulhade, P.Deshpande, “VLSI Design and Analysis of Low Power 6T SRAM Cell Using Cadence Tool” IEEE International Conference on Semiconductor Electronics, 117–121-2008.

56. A. A. Kumar, A. Chalil, “Performance Analysis of 6T SRAM Cell on Planar and FinFET Technology”. IEEE International Conference on Communication and Signal Processing, 0375–0379.2019.

57. M. F. M.Noor, N. E.Alias, A.Hamzah, Z.Johari,M. T. L. Peng “Design of 6T SRAM Cell Using Optimized 20 nm SOI Junctionless Transistor”. IEEE Regional Symposium on Micro and Nanoelectronics, 141–144. 2019.

58. R. M.Premavathi, Q.Tong, K.Choi, Y.Lee, “A Low Power, High Speed FinFET based 6T SRAM Cell with Enhanced Write Ability and Read Stability”.IEEE, International SoC Design Conference, 311–312-2016.

59. O.Thomas, M.Vinet, O.Rozeau, P.Batude, A.Valentian, “Compact 6T SRAM cell with Robust Read/Write Stabilizing Design in 45nm Monolithic 3D IC Technology”. IEEE International Conference on IC Design and Technology, 195–198-2009.

60. J.Prachi, S.C.Neetu, “Performance Evaluation of 6T SRAM Cell Structure and Peripheral Circuitary”, Indian Journal of Applied Research, 3(10), 1–3-2013.

61. Y. Kumar, S.K. Kingra, “Stability Analysis of 6T SRAM Cell at 90nm Technology”, International Journal of Computer Applications, 975, 8887-2016.

62. T.Tripathi, D.S.Chauhan, S.K. Singh, “A Novel Approach to Design SRAM Cells for Low Leakage and Improved Stability”, Journal of Low Power Electronics and Applications, 8(4), 41, 2018b.

63. H. K. Kumar, V.Tomar, “Stability Analysis of Subthreshold 6T SRAM Cell at 45 nm for IoT Application”, International Journal of Recent Technology and Engineering, 8(2)- 2019.

64. S. Saun, H. Kumar, “Design and Performance Analysis of 6T SRAM Cell on Different CMOS Technologies with Stability Characterization” IOP Conference Series: Materials Science and Engineering, 561(1)-2019.

65. S. Kassa, S.Nema, “Energy Efficient Novel Design of Static Random Access Memory Memory Cell in Quantum-dot Cellular Automata Approach” International Journal of Engineering-Transactions B: Applications, 32(5), 720–725-2019.

66. M.Fan.L.V.Hu, Y.N Chen, P.Su, C.T. Chuang, “Analysis of Single-Trap-Induced Random Telegraph Noise on FinFET Devices, 6T SRAM Cell, and Logic Circuits” , IEEE Transactions on Electron Devices, 59(8), 2227–2234-2012.

67. V.Gupta, M.Anis, “Statistical Design of the 6T SRAM Bit Cell”, IEEE Transactions on Circuits and Systems I: Regular Papers, 57(1), 93–104-2009.

68. A.Kranti, C.W Lee, I.Ferain, R.Yan, N.Akhavana, P.Razavi, R.Yu, G.Armstrong, J.P.Colinge, “Junctionless 6T SRAM Cell”, Electronics Letters, 46(22), 1491–1493-2010.

69. X.Deng, T.W.Houston, “Dynamic Threshold Voltage 6T SRAM Cell”. US Patent 6,573,549, 2003.

70. B.R.Huang, F.H.Meng, Y.C.King, C.J.Lin, “Investigation of Parasitic Resistance and Capacitance Effects in Nanoscaled FinFETs and their impact on Static Random-Access Memory Cells”, Japanese Journal of Applied Physics, 56(4S)-2017.

71. S.Jadav, M.Vashisath, “Design and Performance Analysis of Ultra Low Power 6T SRAM Using Adiabatic Technique”, International Journal of VLSI design & Communication Systems, 3(3), 95– 105, 2012.

72. A.Bhaskar, “Design and Analysis of Low Power SRAM Cells”. IEEE Innovations in Power and Advanced Computing Technologies, 1–5-2017.

73. S.Bharti, A.Kumar, R.Kandari, S.Singh, “Performance Analysis of SRAM Cell Designed using MOS and Floating-gate MOS for Ultralow Power Technology”. IEEE, 4th International Conference on Internet of Things: Smart Innovation and Usages, 1–6, 2019.

74. P.Bikki, M.Annapurna, S.Vujwala, “Analysis of Low Power SRAM Design with Leakage Control Techniques”. IEEE, International Conference on Microwave Integrated Circuits, Photonics and Wireless Networks, 400–404, 2019.

75. S.Saxena. S.Mishra, “Analysis of Leakage Current and SNM For 7T SRAM Cell in Nanometre Era”, International Journal of Recent Trends in Engineering & Research, 2(10), 134–141, 2016.

76. P.Narah, S. Nath, “A Comparative Analysis of SRAM Cells in 45nm, 65nm, 90nm Technology”, International Journal of Engineering Research and Application, 8(5), 31–36, 2018.

77. R.Joel, S.S. Gnana, “ Power and Stability Analysis of 6T 7T SRAM Cell Using

Power Gating Techniques", International Journal of Innovative Research in Technology, 3(8), 139–144, 2017.

78. K.F.Sharif , R.Islam, S.N.Biswas "A New Model of High Speed 7T SRAM Cell". IEEE International Conference on Computer, Communication, Chemical, Material and Electronic Engineering, 1–4, 2018.

79. M.Ansari,H. Afzali-Kusha, B.Ebrahimi., Z.Navabi, A. Afzali-Kusha, M.Pedram, "A Near-Threshold 7T SRAM Cell with High Write and Read Margins and Low Write Time for Sub-20 nm FinFET Technologies", Integration, 50, 91–106, 2015.

80. J.Liu, M.B. Clavel, M.K.Hudait, "An Energy-Efficient Tensile-Strained Ge/InGaAs TFET 7T SRAM Cell Architecture for Ultralow- Voltage Applications", IEEE Transactions on Electron Devices, 64(5), 2193– 2200, 2017.

81. K.FSharif, R.Islam, M.Haque, M.A.Keka, S.N.Biswas "7T SRAM Based Memory Cell". IEEE, International Conference on Innovative Mechanisms for Industry Applications, 191–194, 2017.

82. D.Takashima, M.Endo, K.Shimazaki, M.Sai, M.Tanino, "A 7T-SRAM with Data-Write Technique by Capacitive Coupling", IEEE Journal of Solid-State Circuits, 54(2), 596–605, 2018.

83. T.S.Kumar, S.L. Tripathi, "Leakage Reduction in 18 nm FinFET based 7T SRAM Cell using Self Controllable Voltage Level Technique", Wireless Personal Communications, 1–11, 2020.

84. J.K.Mishra, P.K. Misra, M.Goswami, "A Low Power 7T SRAM Cell Using Supply Feedback Technique at 28nm CMOS Technology". IEEE,7th International Conference on Signal Processing and Integrated Networks (SPIN), 597–602, 2020.

85. L. Dilillo, P. Girard, S. Pravossoudovitch and A. Virazel, "Efficient March Test Procedure for Dynamic Read destructive Fault Detection in SRAM memories", Journal of electronic testing: theory and applications, vol. 21, pp. 551-561, 2005.

86. L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel and S. Borri, "March iC: an improved version of March C for ADOFs detection", Proceedings of IEEE VLSI Test Symposium, pp. 129–134, 2004.

87. L. Dilillo, P.Girard, S. Pravossoudovitch, A. Virazel, S. Borri and M. Hage Hassan, "Dynamic read destructive fault in embedded-SRAMs: analysis and March test solution", Proceedings of IEEE Asian Test Symposium, pp. 140–145, 2004.

88. L. Dilillo, P. Rosinger, B. M. Al-Hashimi and P. Girard, “Reducing Power Dissipation in SRAM during Test”, Journal of low power electronics, Vol. 2, No. 2, pp. 271-280, August 2006.

89. L. Dilillo, P. Rosinger, B. M. Al-Hashimi and P. Girard, “Minimizing Test Power in SRAM through Reduction of Pre-charge Activity”, Proceeding of the conference on design automation and test in Europe, Vol. 1, pp. 1-6, March 2006.

90. A. Ney, A. Bosio, L. Dilillo, P. Girard, S. Pravossoudovitch and A. Virazel, “A Signature-based Approach for Diagnosis of Dynamic Faults in SRAMs”, Proceedings of IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era, pp. 1-6, 2008.

91. A. Bosio and G. D. Natale, “March Test BDN:A new March Test for dynamic faults”, Proceedings of IEEE International Conference on Automation, Quality and Testing , Robotics(AQTR), Vol. 1, pp. 85-89, 2008.

92. L. Dilillo and B. M. Al-Hashimi , “March CRF: an Efficient Test for Complex Read Faults in SRAM Memories”, Proceedings of IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems, pp. 1-6, 2007.

93. R. A. Fonseca, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A.Virazel And N. Badereddine, “Impect of Resistive –Bridging Defects in SRAM Core Cell”, proceeding of Fifth IEEE International Symposium on Electronic Design , Test and Application, pp. 265-269, 2010.

94. Luigi Dilillo, Patrick Girard, Serge Pravossoudovitch & Arnoud Virazel , “Resistive-Open Defect influence in SRAM pre-charge circuits: Analysis and characterisation”, Proceeding of the European Test Symposium (ETS’ 05) 2005.

95. Mohamed Tagelsir Mohammadat, Noohul Basheer Zain Ali, Fawnizu Azmadi Hussin, and Mark Zwolinski,“Resistive Open Faults Detectability Analysis and Implications for Testing Low Power Nanometric ICs”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2014.

96. A. Ney, P. Girard, S. Pravossoudovitch, A. Virazel and M. Bastian, “Analysis of Restive- Open Defects in SRAM sense Amplifier”, IEEE Transactions on Very Large scale Integration (VLSI) Systems, Vol.17, pp.1556-1559, 2009.

97. A. Ney, P. Girard, S. Pravossoudovitch, A. Virazel and M. Hage- Hassan, “Unrestored destructive write faults due to resistive-open defects in the write driver of

SRAMs", Proceedings of IEEE VLSI Test Symposium, pp. 361–368, 2007.

98. A. Ney, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel and M. Bastian, "Dynamic two cell incorrect read fault due to resistive-open defects in the sense amplifiers of SRAMs", Proceedings of IEEE European Test Symposium, pp. 97–102, 2007

99. P. Rech, J-M. Galliere, P. Girard, A. Griffoni, J. Boch, F. Wrobel, F. Saigne, and L. Dilillo, "Neutron-Induced Multiple Bit Upsets on Dynamically-Stressed Commercial SRAM Arrays", 12th European Conference on Radiation and Its Effects on Components and Systems (RADECS), 2011.

100. Nihaar Mahatme, Bharat Bhuva,Y-P Fang & Anthony Oates, "Analysis of Multiple Cell Upsets due to Neutrons in SRAMs for a Deep-N- Well process", IEEE International Reliability Physics Symposium, 2011.

101.G. Tsiligiannis, L. Dilillo, A. Bosio, P. Girard, A. Todri, A. Virazel, A.D. Touboul, F. Wrobel & F.Saigné, "Evaluation of Test Algorithms Stress Effect on SRAMs under Neutron Radiation", IEEE 18th International On- Line Testing Symposium (IOLTS), 2012.

102. A. Bosio, L. Dilillo, P. Girard, S. Pravossoudovitch & A. Virazel, "Advanced Test Methods for SRAMs", IEEE 30th VLSI Test Symposium (VTS),2012.

103.S. Borri, M. Hage- Hassan, P. Gijard, S. Pravossoudovitch and A. Virazel, "Defect-Oriented Dynamic fault models for embedded SRAMs", Proceedings of IEEE European test workshop, pp. 23-28, 2003.

104.S. Hamdioui, Z. Al-Ars and A. J. Van de Goor, " Testing static and dynamic faults in random access memories", Proceedings of IEEE VLSI Test Symposium, pp. 395–400, 2002.

105. A. Benso, A. Bosio, S. Di Carlo, G. Di Natale and P. Prinetto, "March AB, March AB1: new March tests for unlinked dynamic memory faults", Proceedings of IEEE International Test Conference, pp. 1–6, 2005.

106. G. Harutunyan, V. A. Vardanian and Y. Zorian , "Minimal March tests for detection of dynamic faults in random access memories", Journal of electronic testing: theory and applications, Vol. 23,No. 1, pp. 55–74, 2007.

107.L. Dilillo, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, M. Bastian and V. Gouin, "Impact of technology scaling on defects and parameter deviations in

embedded SRAMs", Proceedings of IEEE VLSI Test Symposium, pp. 336–336, 2008.
108. Naik S. and Kuwelkar S., "A Novel 8T SRAM with Minimized Power and Delay" IEEE, 2nd International Conference on Recent Trends in Electronics, Information & Communication Technology, 1498– 1501, 2017.
109.Raikwal P., Neema V., and Verma A. "High Speed 8T SRAM Cell Design with Improved Read Stability at 180nm Technology", IEEE, International Conference of Electronics, Communication and Aerospace Technology, 2, 563–568, 2017.
110.Kolsoom M., Ebrahimi B., and Afzali-Kusha A., "A Robust and Low Power 7T SRAM Cell Design", 18th CSI International Symposium on Computer Architecture and Digital Systems, Tehran, 7–8, 2015.
111.Yang B.-D., "A Low-Power SRAM Using Bit-Line Charge- Recycling for Read and Write Operations", IEEE Journal of Solid-State Circuits, 45(10), 2173–2183,2010.
112.A. B. Kahng, "Machine Learning Applications in Physical Design: Recent Results and Directions", Statistical and Machine Learning-Based CAD, March 25–28, 2018.
113.B.Shook, P.Bhansali, C. Kashyap, C. Amin, S.Joshi, "MLParest: Machine Learning based Parasitic Estimation for Custom Circuit Design" IEEE-2020.
114.S.K.Samal, G.Chen, S. K. Lim "Machine Learning Based Variation Modeling and Optimization for 3D ICs", Journal of information and Communication convergence Engineering, 14(4): 258-267, Dec. 2016.
115.P.Princy , N.M.Sivamangai "An Efficient Wavelet Based Transient Current Test towards Detection of Data Retention Faults in SRAM", Journal of Electronic Testing, Springer-2019.
116.Rob Decker, Frans Beenker, and Loek Thijssen, "A Realistic Fault Model and Test Algorithms for Static Random Access Memories", IEEE transactions on Computer Aided Design, Vol-9, 1990, pp:567-572.
117. M.H. Abu-Rahma and M. Anis, "Variability in nano meter technologies and Impact on SRAM", Springer Media, New yark 2013.
118. Said Hamdioui, Ad J. van de Goor, "An Experimental analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests", IEEE, pp:131-138.
119. M.Venkatesham, S.K.Sinha and M.Parvathi, V.Sharma, "Comparative Analysis of Open and Short Defects in embedded SRAM using Parasitic Extraction Method for Deep Submicron Technology" Wireless Personal Communications (2023)

132:2123–2141.

120. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu "A Parasitic Extraction Method for Realistic Linked Faults in Two Cell SRAMs" 5th International Conference on Intelligent Circuits and Systems (ICICS-), October, 2023.

121. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Fault Detection and Analysis in embedded SRAM for sub nanometer technology", IEEE International Conference on Applied Artificial Intelligence and Computing(ICAAIC), Salem, India, May, 2022.

122. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Study on Paradigm of Variabl Length SRAM Embedded Memory Testing", 5th IEEE International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, Dec-2021.

123. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Analysis of Open Defect Faults in Single 6T SRAM Cell Using R and C Parasitic Extraction Method", IEEE International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bangaluru, India, Nov-2021.

124. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Extraction of Undetectable Faults in 6T- SRAM Cell". IEEE International Conference on Communication, Control and Information Sciences (ICCISc), Idukki, India, June-2021.

125. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Open Defect Fault Analysis in Single Cell SRAM Using R and C Parasitic Extraction Method", International Conference on Information and Communication Engineering (ICICE), Hyderabad, India, Feb-2021.

126. N.C. Yang, "Long Short-Term Memory-Based Feedforward Neural Network Algorithm for Photovoltaic Fault Detection Under Irradiance Conditions", IEEE Transactions on Instrumentation And Measurement, VOL. 73, 2024.

127. C Fan, K Xiahou, L Wang, Q. H. Wu, "Data-driven Fault Detection of Multiple Open-circuit Faults for MMC Systems Based on Long Short-term Memory Networks", CSEE

Journal of Power And Energy Systems, VOL. 10, NO. 4, JULY 2024.

128. T Wu, W Fan, Y Gu , F Fan, Q Li, “A New Test Algorithm and Fault Simulator of Simplified Three-Cell Coupling Faults for Random Access Memories”, IEEE, August-2024.

129. A. A. Khan, S. Ollivier, F.Hameed, J. Castrillon, A.K. Jones, “DownShift: Tuning Shift Reduction with Reliability for Racetrack Memories”, IEEE Transactions on Computers, VOL. 72, NO. 9, September 2023.

130. B.Roy , S. Adhikari, S.Datta, K.J. Devi, A.D.Devi, F.Alsaif, S.Alsulamy, T.S.Ustun “Deep Learning Based Relay for Online Fault Detection, Classification, and Fault Location in a Grid-Connected Microgrid”, IEEE, June-2023.

131. S.Taghipour, M.Kama,R.N.Asli, A.A.Kusha, M. Pedram, “LCHC-DFT: A Low-Cost High-Coverage Design-for-Testability Technique to Detect Hard-to-Detect Faults in STT-MRAMs in the Presence of Process Variations”, Ieee Transactions On Device And Materials Reliability, VOL. 22, NO. 4, DEC 2022.

132. A. Swetapadma, S. Chakrabarti, A.Y. Abdelaziz, H.H.Alhelou, “A Novel Relaying Scheme Using Long Short Term Memory for Bipolar High Voltage Direct Current Transmission Lines”, IEEE August-2021.

133. D.Han, H. Lee, S.Kang, “Effective Spare Line Allocation Built-in Redundancy Analysis With Base Common Spare for Yield Improvement of 3D Memory”, IEEE, May-2021.

134. C.Xie, X.Li,Y.Lei, H. Chen,Q. Wang, J. Guo, J.Miao, Y.Lv, Z. Song, “BIST-Based Fault Diagnosis for PCM With Enhanced Test Scheme and Fault-Free Region Finding Algorithm”, IEEE Transactions on Very Large Scale Integration (Vlsi) Systems, VOL. 28, NO. 7, July 2020.

135. M.Soltani, M. Kamal , A.A.Kusha, M.Pedram, “RandShift: An Energy-Efficient Fault-Tolerant Method in Secure Nonvolatile Main Memory”, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, VOL. 28, NO. 1, January 2020.

LIST OF PUBLICATIONS

Venkatesham Maddela, Sanjeet Kumar Sinha, Parvathi Muddapu and Vinay Sharma "Comparative Analysis of Open and Short Defects In Embedded SRAM Using Parasitic Extraction Method for Deep Submicron Technology", Springer, Wireless Personal Communications, 132:2123–2141, 2023.(SCI)

https://link.springer.com/article/10.1007/s11277-023-10704-w

Under Review

Venkatesham Maddela, Sanjeet Kumar Sinha, Parvathi Muddapu a "Parasitic RC Estimation and Defect Prediction using Machine Learning", Springer, Analog Integrated Circuits and Signal Processing.

LIST OF CONFERENCES

1. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu "A Parasitic Extraction Method for Realistic Linked Faults in Two Cell SRAMs" 5th International Conference on Intelligent Circuits and Systems (ICICS-2023), October 12-13th, 2023.

2. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Fault Detection and Analysis in embedded SRAM for sub nanometer technology", IEEE International Conference on Applied Artificial Intelligence and Computing(ICAAIC 2022), Salem, India, May 9 to 11, 2022.
https://ieeexplore.ieee.org/document/9793265

3. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Study on Paradigm of Variabl Length SRAM Embedded Memory Testing", 5th IEEE International Conference on Electronics, Communication and Aerospace Technology (ICECA 2021), Coimbatore, India, Dec. 2 to 4, 2021.
https://ieeexplore.ieee.org/document/9675983

4. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Analysis of Open Defect Faults in Single 6T SRAM Cell Using R and C Parasitic Extraction Method", IEEE International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON-2021), Bangaluru, India, Nov. 19 to 21, 2021.
https://ieeexplore.ieee.org/document/9687916

5. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Extraction of Undetectable Faults in 6T- SRAM Cell". IEEE International Conference on Communication, Control and Information Sciences (ICCISc-2021), Idukki, India, June 16 to18, 2021.
https://ieeexplore.ieee.org/document/9484987

6. Venkatesham Maddela, Sanjeet Kumar Sinha and Parvathi Muddapu, "Open Defect Fault Analysis in Single Cell SRAM Using R and C Parasitic Extraction Method", International Conference on Information and Communication Engineering (ICICE), Hyderabad, India, Feb. 11 to 13, 2021.
https://ieeexplore.ieee.org/document/9687916

LIST OF WORKSHOPS ATTENDED

1. One week FDP on "Machine Learning and its Application" 20 th March – 1 St April 2022

2. Online Microwind workshop on "Recent trends in FinFET & Nano-Sheet Cell Design", 20-21 August 2021

3. AICTE Training and Learning (ATAL) FDP on "Machine Learning Techniques in VLSI Design", 26-30 July 2021.

4. One-week national level FDP on "Recent Trends in VLSI Design", 19 -23 July 2021.

5. Online Microwind workshop on "FinFET & Nano-Sheet Cell Design, Now & Road ahead", 11-12 September 2022

6. Online Microwind workshop on "FinFET, Nano-Sheet Cell Design, Now & Road ahead", 29-31 August 2023