Yield-Aware Leakage Power Reduction of On-Chip SRAMs

Afshin Nourivand

A Thesis
In the Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Electrical and Computer Engineering) at
Concordia Uuniversity
Montréal, Québec, Canada

October 2010

©Afshin Nourivand, 2010

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Afshin Nourivand**

Entitled: **Yield-Aware Leakage Power Reduction of On-Chip SRAMs**

and submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of the university and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. M. Paraschivoiu

______________________________ External Examiner
Dr. M. Sachdev

______________________________ External to Program
Dr. T. Radhakrishnan

______________________________ Examiner
Dr. G. Cowan

______________________________ Examiner
Dr. M. Z. Kabir

______________________________ Thesis Co-Supervisor
Dr. A. J. Al-Khalili

______________________________ Thesis Co-Supervisor
Dr. Y. Savaria

Approved by ______________________________
Dr. M. Kahrizi, Graduate Program Director

September 29, 2010 ______________________________
Dr. Robin A. L. Drew, Dean
Faculty of Engineering and Computer Science

Abstract

Yield-Aware Leakage Power Reduction of On-Chip SRAMs

Afshin Nourivand, Ph.D.
Concordia University, 2010

Leakage power dissipation of on-chip static random access memories (SRAMs) constitutes a significant fraction of the total chip power consumption in state-of-the-art microprocessors and system-on-chips (SoCs). Scaling the supply voltage of SRAMs during idle periods is a simple yet effective technique to reduce their leakage power consumption. However, supply voltage scaling also results in the degradation of the cells' robustness, and thus reduces their capability to retain data reliably. This is particularly resulting in the failure of an increasing number of cells that are already weakened by excessive process parameters variations and/or manufacturing imperfections in nano-meter technologies. Thus, with technology scaling, it is becoming increasingly challenging to maintain the yield while attempting to reduce the leakage power of SRAMs. This research focuses on characterizing the yield-leakage tradeoffs and developing novel techniques for a yield-aware leakage power reduction of SRAMs.

We first demonstrate that new fault behaviors emerge with the introduction of a low-leakage standby mode to SRAMs. In particular, it is shown that there are some types of defects in SRAM cells that start to cause failures only when the drowsy mode is activated. These defects are not sensitized in the active operating mode, and thus escape the traditional March tests. Fault models for these newly observed fault behaviors are developed and described in this thesis. Then, a new low-complexity test algorithm, called March RAD, is proposed that is capable of detecting all the drowsy faults as well as the simple traditional faults.

Extreme process parameters variations can also result in SRAM cells with very weak data-retention capability. The probability of such cells may be very rare in small

memory arrays, however, in large arrays, their probability is magnified by the huge number of bit-cells integrated on a single chip. Hence, it is critical also to account for such extremal events while attempting to scale the supply voltage of SRAMs. To estimate the statistics of such rare events within a reasonable computational time, we have employed concepts from extreme value theory (EVT). This has enabled us to accurately model the tail of the cell failure probability distribution versus the supply voltage. Analytical models are then developed to characterize the yield-leakage tradeoffs in large modern SRAMs. It is shown that even a moderate scaling of the supply voltage of large SRAMs can potentially result in significant yield losses, especially in processes with highly fluctuating parameters. Thus, we have investigated the application of fault-tolerance techniques for a more efficient leakage reduction of SRAMs. These techniques allow for a more aggressive voltage scaling by providing tolerance to the failures that might occur during the sleep mode. The results show that in a 45-nm technology, assuming 10% variation in transistors threshold voltage, repairing a 64KB memory using only 8 redundant rows or incorporating single error correcting codes (ECCs) allows for $\sim$ 90% leakage reduction while incurring only $\sim$ 1% yield loss. The combination of redundancy and ECC, however, allows to reach the practical limits of leakage reduction in the analyzed benchmark, i.e., $\sim$ 95%.

Applying an identical standby voltage to all dies, regardless of their specific process parameters variations, can result in too many cell failures in some dies with heavily skewed process parameters, so that they may no longer be salvageable by the employed fault-tolerance techniques. To compensate for the inter-die variations, we have proposed to tune the standby voltage of each individual die to its corresponding minimum level, after manufacturing. A test algorithm is presented that can be used to identify the minimum applicable standby voltage to each individual memory die. A possible implementation of the proposed tuning technique is also demonstrated. Simulation results in a 45-nm predictive technology show that tuning standby voltage of

SRAMs can enhance data-retention yield by an additional $10\% - 50\%$, depending on the severity of the variations.

Acknowledgements

I would like to express my sincere gratitude and appreciation to my academic supervisors, Professor Asim J. Al-Khalili and Professor Yvon Savaria, for their insightful guidance and generous support throughout this research. They both have been instrumental in the success of this research. Professor Al-Khalili was always there to give me the moral support and encouragement during tough times. His sound vision, knowledge, and experience helped me improve both professionally and personally. I am also deeply indebted and hold great respect for my co-supervisor, Professor Savaria, whom without his guidance and support this work would not have been possible. His expertise and critical suggestions have shaped the character of this research. I continue to be amazed by the breadth of his knowledge, his innovative ideas, and his endless energy for research. Indeed, I feel truly privileged for having this unique experience of working under the supervision of such great professors.

I want to thank my thesis defense committee members: Dr. Manoj Sachdev, Dr. Glenn Cowan, Dr. Zahangir Kabir, and Dr. Thiruvengadam Radhakrishnan for their helpful comments and suggestions. My very special thanks go to my external examiner, Dr. Manoj Sachdev, for the time he took to carefully read and comment on my dissertation. His insightful comments and suggestions helped me enhance this work significantly.

I would also like to thank those people in the ECE department who provided a positive environment that encouraged me to aim at high-quality research. Namely, I am grateful to Dr. Omair Ahmad for his support during the first 2 years of my studies. I also thank Dr. Otmane Ait Mohamed for being on my qualification exam committee. I would also like to thank Ted Obuchowicz for keeping the CAD tools running. My special thanks also go to Pamela Fox, the Ph.D. program coordinator,

and Kimberly Adams, the TA coordinator. I also want to thank all former and current members of the VLSI group for the cheerful moments that we shared in the lab.

Last but surely not least, I am truly grateful and indebted to my parents, for their endless sacrifices and for teaching me the fundamental principles of uprightness, honesty, compassion and hard work.

To my beloved parents.

Contents

Contents viii

List of Figures xii

List of Tables xv

Acronyms xvi

1 Introduction 1

1.1 Motivation 5

1.2 Contributions and Proposed Solutions 6

1.2.1 New Fault Behaviors and Their Impact on Low-Leakage SRAMs 6

1.2.2 Modeling the Yield-Leakage Tradeoff in Large SRAM Arrays Considering Extreme Failure Events 7

1.2.3 Aggressive Leakage Reduction of SRAMs Using Fault-Tolerance Techniques 8

1.2.4 Post-Silicon Tuning of Standby Supply Voltage for Reduction of Parametric Yield Losses Due to Data-retention Failures . . 8

1.3 Organization of the Dissertation 9

2 Background **11**

2.1 SRAMs Organization and Operation 11

2.1.1 SRAMs Organization 11

2.1.2 SRAM Cells 13

2.2 Leakage Power Dissipation in SRAMs 14

2.2.1 Subthreshold Current 15

2.2.2 Gate-Tunneling Current 15

2.3 SRAM Leakage Reduction Techniques 16

2.3.1 Supply Voltage Scaling 16

2.3.2 Source Biasing 18
2.3.3 Architectural Level Leakage Reduction Techniques 19
2.4 Yield Losses Due to the Introduction of a Drowsy Mode to SRAMs . 24
2.4.1 Impact of Process Variations on Drowsy SRAMs 26
2.4.2 Impact of Defects on Drowsy SRAMs 29
2.4.3 Importance of Extremal Events in Large SRAMs 31
2.4.4 Yield-Leakage Tradeoff in SRAMs 32
2.5 Summary 32

3 New Fault Models and Their Impact on Low Leakage Drowsy SRAMs 34

3.1 Impact of Defects on Drowsy SRAMs 35
3.1.1 Data Retention Voltage (DRV) of Defective Cells 36
3.1.2 Wake-up Time 38
3.2 Simulation Methodology 39
3.3 Fault Modeling and Notation 41
3.3.1 Open Defects in SRAM Cells 42
3.3.2 Functional Fault Models 43
3.3.3 Fault Notation 44
3.4 SRAM Drowsy Faults Due to Resistive-Open Defects 45
3.4.1 Static Drowsy Faults (SDF) 47
3.4.2 Dynamic Drowsy Faults (DDF) 50
3.5 Testing for Drowsy Faults 54
3.5.1 March RAD 55
3.5.2 Test Implications of Drowsy Cache Architectures 57
3.5.3 Sensitivity to Process Parameters Variations 58
3.6 Summary 60

4 Aggressive Leakage Reduction of SRAMs Using Fault-Tolerance Techniques: The Yield-Power Tradeoff 61

4.1 Maximum Applicable Source-Bias Voltage to a Memory 62
4.2 Modeling the Tail of the V_{SBmax} Distribution 64
4.2.1 Simulation Setup and Process Variation Model 66
4.2.2 Tail Modeling Procedure 67

4.3 Computing Array Yield at Elevated V_{SB} 68
4.3.1 Cell Failure Probability at Elevated V_{SB} 69
4.3.2 Array Failure Probability at Elevated V_{SB} 70
4.4 Estimating Net Power Savings 74
4.4.1 Overhead Power Associated with Fault-Tolerance Techniques . 74
4.4.2 Estimating Leakage Power Considering Process Variations . . 75
4.5 Simulation Results and Discussion 75
4.5.1 Yield Degradations due to Source-Biasing 76
4.5.2 Yield-Leakage Tradeoff Using Different Fault-Tolerance Techniques 79
4.6 Summary 82
5 Post-Silicon Tuning of Standby Supply Voltage in SRAMs to Reduce Parametric Data-Retention Failures 83
5.1 Inter-Die Distribution of V_{DDLmin} 84
5.2 Minimum Applicable Standby Voltage to a Memory Die (V_{DDLmin}) . 85
5.2.1 Joint Impact of Inter- and Intra-Die Variations on V_{DDLmin} of SRAMs 87
5.2.2 Impact of the Size of Memory on its V_{DDLmin} 87
5.2.3 Impact of Adding Redundancy on V_{DDLmin} of SRAMs 87
5.2.4 Mathematical Model of Inter-Die V_{DDLmin} Distribution 88
5.2.5 Tradeoff between Leakage Reduction and Yield of SRAMs . . 89
5.3 Estimating Data-Retention Failure Probability as a Function of Supply Voltage 90
5.3.1 Estimation of Rare Failure Events 90
5.3.2 Simulation Methodology 91
5.4 Computing Array Yield from Cell Failure Probability 97
5.4.1 Yield of a Memory Without Redundancy 97
5.4.2 Yield of a Memory With Redundancy 98
5.4.3 Poisson Yield Model 99
5.5 Applying an Identical Standby Voltage to All Dies 100
5.5.1 Impact of Inter-die Variations on Data-Retention Failure Probability 101
5.5.2 Impact of Inter-die Variations on Array Failure Probability . . 103

5.5.3 Impact of Inter-die Variations on Data-Retention Yield 104
5.5.4 Impact of Process Parameters Variations on Leakage Yield . . 106
5.5.5 Tradeoff Between Data-Retention and Leakage Yield 107
5.6 Yield Enhancement by Standby Supply Voltage Tuning 108
5.6.1 Post-Silicon Standby Voltage Tuning 108
5.6.2 Overhead of the Tuning Technique 112
5.7 Simulation Results for Yield Enhancements and Discussions 113
5.7.1 V_{DDLmin} Distribution 114
5.7.2 Yield Enhancements by Standby Voltage Tuning 115
5.7.3 Yield Losses Due to Dies With Excess Leakage in Case of Voltage Tuning 117
5.7.4 Uncorrelated Inter-Die Shift for NMOS and PMOS 117
5.8 Summary 118
6 Conclusions and Future Work **119**
6.1 Contributions and Main Results 120
6.2 Future Work 122
6.2.1 More Efficient Tests for Detection of Drowsy Faults 122
6.2.2 A Built-In Technique for Self-Tuning of Standby Supply Voltage Against Run-Time Variations 123
6.2.3 Compensating for Systematic Intra-Die Variations 124
References **125**

List of Figures

1.1 (a) 24MB of on-chip L3 cache in Intel's 8-core Xeon processor. Adapted from [4] (Copyright 2010 IEEE) and (b) projections of logic/memory composition of low-power SOC designs [5]. 2

1.2 Increasing leakage power fraction of total processor power consumption with technology scaling [6]. 3

1.3 (a) Random dopant fluctuations, adapted from [7] (Copyright 2008 Intel) and (b) scaling trend of threshold voltage variation [8, 9, 5]. . . 4

1.4 Examples of weak open defects: (a) cross section of a metal open line, the metal cavity and formation of a weak open due to the Ti barrier, and (b) a resistive via. Adapted from [11] (Copyright 2002 IEEE). . . 5

2.1 A typical SRAM organization. 12

2.2 Conventional 6T SRAM cell. 13

2.3 SRAM cell leakage currents during standby mode. The leaking transistors are shown in dotted lines. 15

2.4 (a) SRAM cell leakage currents at reduced supply voltage and (b) circuit simulation results for leakage currents at reduced supply voltages for an SRAM cell in a 45-nm technology at $T = 27°C$. 17

2.5 SRAM supply scaling by power gating. The programmable bias transistors enable controlling of the virtual VDD. 18

2.6 (a) SRAM cell leakage currents at raised source-line voltage and (b) circuit simulation results for leakage currents at raised source-line voltages for an SRAM cell in a 45-nm technology at $T = 27°C$. 19

2.7 SRAM ground gating. The programmable bias transistors enable to control the virtual GND. 20

2.8 DRG-Cache. 21

2.9 Leakage reduction technique using periodic sleep policy. 22

2.10 Leakage reduction with wake-up counters. . . . 23

2.11 (a) Butterfly curve of a balanced cell at different supply voltages. SNM of the cell is reduced to zero at $V_{DD} = 200mV$, and (b) Waveforms for the voltage of storage nodes, i.e., T and F, of a balanced cell as the supply voltage is reduced down to zero. . . . 25

2.12 Inter and intra die variation modeling. . . . 27

2.13 Random within-die variations in threshold voltage in 65-nm and 45-nm technologies. Adapted from [7] (Copyright 2008 Intel). . . . 28

2.14 (a) Butterfly curves of an imbalanced cell at different supply voltages. SNM_{low} is reduced to zero before SNM_{high} at $V_{DD} = 270mV$. and (b) waveforms for the voltage of storage nodes, i.e., T and F, of an imbalanced cell as the supply voltage is reduced down to zero. . . . 29

2.15 Histogram of the DRV from a 5000 point Monte Carlo simulation of SRAM cells in a 45-nm predictive technology node. . . . 30

2.16 (a) A resistive open defect in pull-up path of a 6T SRAM cell, (b) simulation results showing the reduction of DRV in defective SRAM cells. . . . 31

3.1 (a) A resistive open defect in pull-up path of a 6T SRAM cell, (b) Shmoo plot showing pass/fail status of an SRAM cell for two parameters: i) resistance of open defect and ii) standby voltage. . . . 37

3.2 Simulation results showing the difference in wake-up time of a healthy cell and a defective cell. . . . 39

3.3 Simulation setup. . . . 41

3.4 All possible open defects in a 6T SRAM cell. . . . 43

3.5 HSPICE simulation results of a defective cell exhibiting (a) a drowsy data-retention fault (DDRF) and (b) a drowsy transition fault (DTF). 48

3.6 (a) Simulation results of a defective cell exhibiting (a) a drowsy read-destructive fault (DRDF) and (b) a drowsy incorrect read fault (DIRF). 51

3.7 (a) Monte Carlo simulation results of (a) a healthy cell and (b) a defective cell with $R_{OC1} = 40M\Omega$, when the March RAD test is performed on them. . . . 59

4.1 Histogram of maximum applicable source-bias voltage to SRAM cells (V_{SBmax}) obtained by 5000 Monte Carlo simulations. . . . 63

4.2 Fitted GPD to the tail of V_{SBmax} distribution ($\sigma Vt/Vt = 10\%$). . . . 69

4.3 Cell failure probability at elevated source-bias voltages up to the first percentile point. ($\sigma Vt/Vt = 10\%$). . . . 70

4.4 Organization of a typical on-chip cache. Leakage reduction is applied only to the data array. 71

4.5 Yield of the 64KB SRAM as a function of the source-bias voltage (a) at different levels of process variations when no fault-tolerance technique is present, (b) when $R = 8$, $R = 16$, or $R = 32$ redundant rows are added, (c) when SEC-DED or DEC-TED codes are employed, (d) when $R = 8$ redundant rows in combination with a SEC-DED code are employed. 77

4.6 The yield-leakage tradeoff in SRAMs: the leakage reductions and yield losses in a 64KB memory as the source-bias voltage is raised. 80

4.7 Feasible reduced leakage of a 64KB memory using various fault-tolerance techniques subject to a 99% target yield. (100% leakage means no reduction is possible.) 81

5.1 A conceptual illustration of the long tail V_{DDLmin} distribution and probability of functional array versus standby supply voltage. 84

5.2 Histogram of the DRV from a 5000 point Monte Carlo simulation of SRAM cells in a 45-nm predictive technology node. 86

5.3 Inter and intra-die variation modeling. 93

5.4 (a) Probability of DRFs versus standby voltage at two different inter-die corners. 101

5.5 Probability of DRFs at various inter-die corners for $V_{DDL} = 0.5V$ and $V_{DDL} = 0.3V$. 102

5.6 Array failure probability of a 1Mb memory versus standby voltage with different levels of available redundancy ($r = R/M$) at (a) nominal-Vt corner, and (b) $\Delta Vt_{inter} = -100mV$. 104

5.7 Dies at some skewed process points become non-repairable by the available redundancy ($r = 1\%$), when an identical standby voltage is applied to all dies. 105

5.8 Data-retention and leakage yield versus standby voltage. 107

5.9 Standby supply voltage tuning scheme. 111

5.10 (a) Distribution of V_{DDLmin} at various inter-die corners, and (b) Distribution of V_{DDLmin} for memory dies in a process with $\sigma_{Vt_{inter}} = 10\%$. 114

5.11 (a) Yield of a 1Mb memory versus relative standard deviation (RSD) of inter-die Vt variation at fixed and tuned standby voltages with no redundancy, and (b) with 1% redundancy ratio. 115

List of Tables

3.1 Single-Cell Static and Dynamic Faults In Drowsy SRAM Due To PODs. 46
3.2 March RAD Test. 55
3.3 March RAD Fault Coverage. 57

4.1 The organization of the simulated memory array. 76

Acronyms

BIST	Built-In Self Test
CDF	Cumulative Distribution Function
CMOS	Complementary Metal-Oxide-Semiconductor
DDRDF	Drowsy Deceptive Read Destructive Fault
DDRF	Drowsy Data-Retention Fault
DEC-TED	Double Error Correcting-Triple Error Detecting
DIRF	Drowsy Incorrect Read Fault
DPM	Defects Per Million
DRDF	Drowsy Read Destructive Fault
DRF	Data-Retention Fault
DRV	Data-Retention Voltage
DTF	Drowsy Transition Fault
DUSF	Drowsy Undefined State Fault
ECC	Error Correcting Code

EVT	Extreme Value Theory
FFM	Functional Fault Model
FP	Fault Primitive
GEV	Generalized Extreme Value
GPD	Generalized Pareto Distribution
IS	Importance Sampling
ITRS	International Technology Roadmap for Semiconductors
MC	Monte Carlo
MLE	Maximum Likelihood Estimation
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MPSoC	Multi-Processor System-on-a-Cchip
MTTF	Mean Time To Failure
NMOS	N-channel MOSFET
OSBV	Optimum Source-Bias Voltage
PDF	Probability Density Function
PMOS	P-channel MOSFET
POD	PMOS Open Defect
PTM	Predictive Technology Model
PWM	Probability Weighted Moments
RDF	Random Dopant Fluctuation

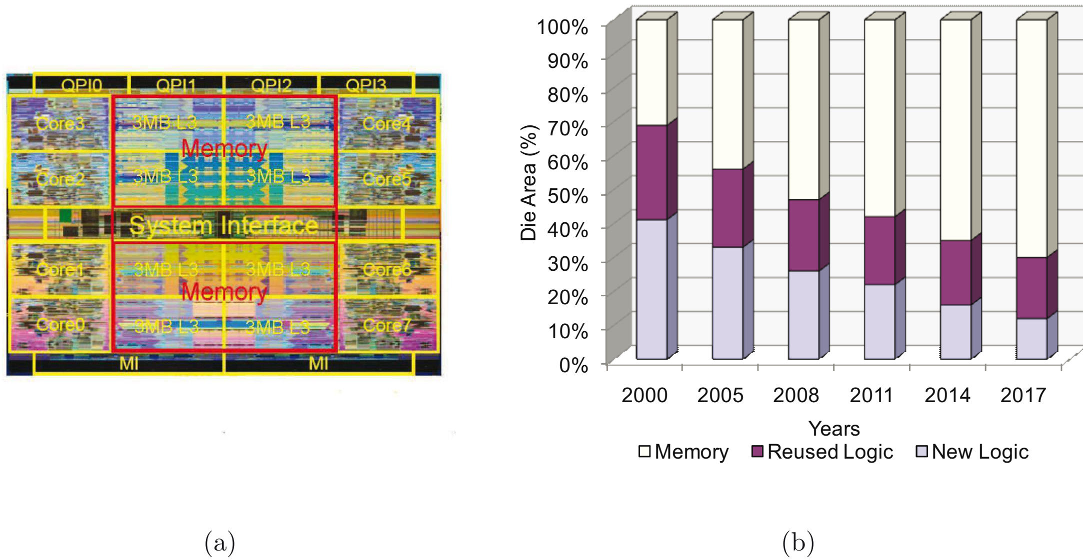
RSD	Relative Standard Deviation
SD	Spot Defect
SDRF	Standby Data Retention Fault
SEC-DED	Single Error Correcting-Double Error Detecting
sF	strong Fault
SNM	Static Noise Margin
SoC	System-on-a-Chip
SRAM	Static Random Access Memory
wF	weak Fault

Chapter 1

Introduction

Aggressive scaling of CMOS devices in the last four decades has enabled the semiconductor industry to meet its ever-increasing demand for higher performance and higher integration densities. However, this trend is encountering several major challenges in the nano-meter era, due to the high integration levels as well as the physical limitations of semiconductor devices. High power consumption is one of the major challenges of integrated circuit design in nano-scale technologies [1]. For high-performance applications, large power dissipations within a small die area are resulting in alarming temperatures, posing serious reliability concerns. For battery operated devices, on the other hand, increased power consumption is drastically limiting the battery lifetime.

Embedding memory into the dies is proven to be a very effective way to improve the performance of systems while reducing their overall power consumption [1, 2, 3]. On-chip cache memory plays a major role in the enhancement of the performance of microprocessors by providing a higher bandwidth and lower latency, while consuming much less power compared to logic. As a result, increasingly larger fractions of chip area are being dedicated to on-chip memories in state-of-the-art microprocessors and



(a)

(b)

Figure 1.1. (a) 24MB of on-chip L3 cache in Intel's 8-core Xeon processor. Adapted from [4] (Copyright 2010 IEEE) and (b) projections of logic/memory composition of low-power SOC designs [5].

system-on-chips (SoCs). For example, the latest 8-core Xeon® processor from Intel® contains 24MB of on-chip L3 (level 3) cache [4], that occupies the majority of the die area (see Figure 1.1(a)). It is predicted that this trend will continue in the future technologies as shown in Figure 1.1(b), where in 2017, more than 70% of the die area will be occupied by memory [5].

Being the largest block on the chip, a low power robust memory design is crucial for the overall reliability, yield and power of the SoCs. There are various design options to realize embedded memories [2]. Currently, static random access memory (SRAM) is the most popular choice for high performance designs, mainly due to its fast access time and compatibility with the mainstream CMOS bulk technology [1, 2]. With scaling to sub-100nm regime, satisfying the multi-dimensional requirements of low power, high yield and reliability of SRAMs has become increasingly difficult, due to the generally conflicting nature of these requirements [3]. Some of the major challenges of SRAM design are as follows:

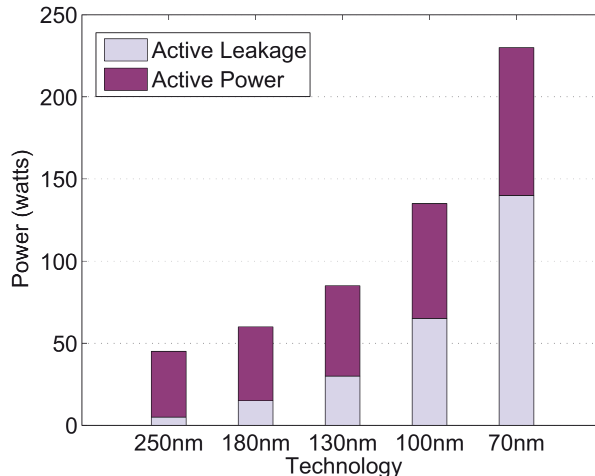
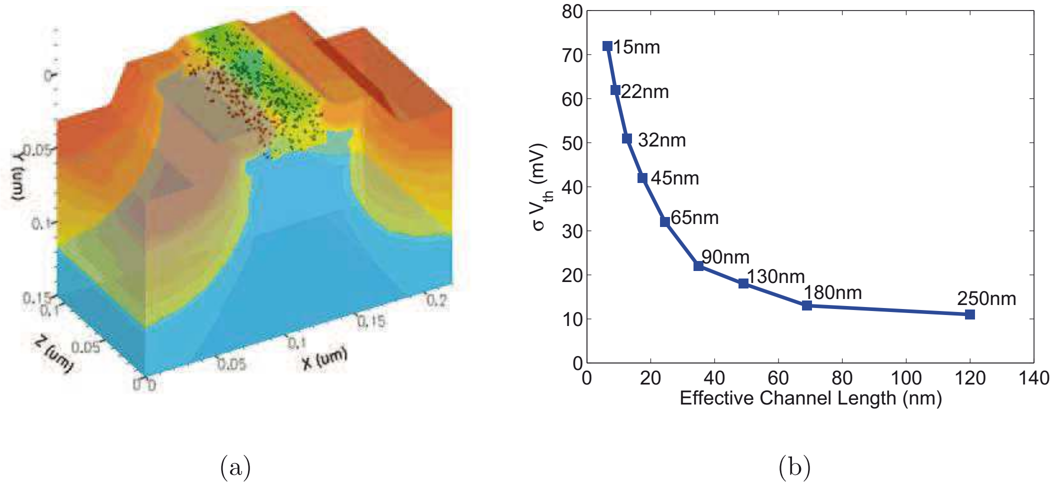


Figure 1.2. Increasing leakage power fraction of total processor power consumption with technology scaling [6].

Leakage power: With technology scaling, transistors exhibit larger leakage currents [1], and as a result, the leakage power consumption in microprocessors and SoCs has started to dominate the total chip power consumption (see Figure 1.2) [6]. A significant fraction of the chips' leakage power is dissipated by SRAMs, as they must remain powered on all the time to retain their data, while their large number of transistors constantly draw leakage power [1]. A low leakage operation of SRAMs is particularly critical for portable devices, as they spend most of their battery lifetime in standby mode.

Process parameters variations: As process geometries continue to shrink, controlling the variations in device parameters during fabrication is becoming increasingly difficult [10, 7]. Random variations, e.g., random dopant fluctuations (RDF) (see Figure 1.3(a)), are particularly troublesome as they are unpredictable, and thus, despite the systematic variations, they cannot be minimized by design-time techniques. The intrinsic random variations are inversely proportional to the gate area, and thus their impact on device parameters, e.g., threshold voltage, are significantly increasing with technology scaling [8, 9] (see Figure 1.3(b)). SRAMs, in particular, are profoundly



(a) (b)

Figure 1.3. (a) Random dopant fluctuations, adapted from [7] (Copyright 2008 Intel) and (b) scaling trend of threshold voltage variation [8, 9, 5].

impacted by random variations as they use minimum-size transistors to obtain higher integration densities [1, 3]. The process parameters variations translate into fluctuations in SRAM metrics such as minimum operating voltage, access time, etc. Modern embedded SRAMs contain millions of transistors, thus some cells will necessarily exhibit behavior far out in the tail of the metrics distribution (as far as 6-7σ) [1]. Such extreme cases can easily fall out of the design specifications and cause failures. Thus, to maintain a sufficient yield in scaled technologies, it is imperative to effectively deal with the process variation issues in SRAMs.

Manufacturing defects: Due to manufacturing inaccuracies, spots of extra, missing or undesired material can cause undesired shorts or opens in circuits. With the increasing complexity of processes and the large number of interconnect layers, the probability of these defects is increasing with technology scaling. Traditionally, fault models have been developed to describe the behavior of SRAMs, in the presence of such defects, during normal operating modes. However, new operating modes, e.g., sleep mode, are being introduced to SRAMs in modern integrated circuits. The introduction of these new operating modes to SRAMs can cause new faulty behaviors to

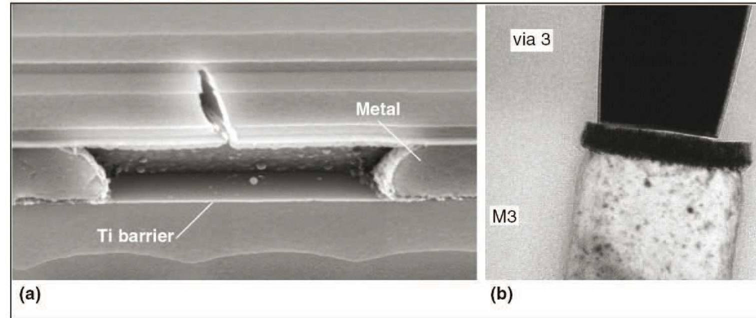


Figure 1.4. Examples of weak open defects: (a) cross section of a metal open line, the metal cavity and formation of a weak open due to the Ti barrier, and (b) a resistive via. Adapted from [11] (Copyright 2002 IEEE).

emerge. Therefore, there can be defects in SRAM cells that while not causing a malfunction during the normal operating mode, start to do so when memory is switched to other operating modes. Examples of such defects can be weak opens or shorts that connect nodes weakly by having a finite parasitic resistance (see Figure 1.4) [11]. Such weak defects can deteriorate various metrics of SRAM cells without causing a hard failure in normal operating conditions. However, they can turn into a strong fault at deteriorated or low power operating conditions such as the reduced supply voltage during a sleep mode. Thus, it is crucial to test memories in all operating modes, in order to minimize the number of defective parts.

1.1 Motivation

As mentioned above, due to the dedication of a significant fraction of the chip area to SRAMs in modern microprocessors and SoCs, their low power dissipation, high yield and high reliability are crucial for the overall success of the designs. Unfortunately, fulfilling the joint requirements of low power dissipation and high yield/reliability of SRAMs poses a "design paradox". Turning the circuits knobs to reduce the leakage power consumption of SRAMs also results in the reduction

of the cells robustness, making them vulnerable to parametric data-retention failures (DRF)s. The failure rates are accentuated in new technologies due to the ever-increasing process parameters variations and manufacturing imperfections. Moreover, due to the sheer number of data cells in contemporary on-chip memories, even a very small failure probability can translate to significant yield losses. Thus, it is crucial to maintain the correct operation across the entire array, while trying to reduce the power dissipation of SRAMs. In this thesis, we have made an attempt to address the contradictory design requirements of joint low-power dissipation and high yield in SRAMs, and propose solutions for their yield-aware leakage power reduction.

1.2 Contributions and Proposed Solutions

1.2.1 New Fault Behaviors and Their Impact on Low-Leakage SRAMs

We have demonstrated that there are faults, not sensitized in normal operation, that appear when an SRAM is switched to a low-leakage drowsy operating mode. Fault models for these newly observed fault behaviors are developed and described in this thesis. Based on the derived fault models, a new low-complexity test algorithm, called March RAD, is proposed, that is capable of detecting all the drowsy faults as well as the traditional simple faults.

It is also shown that as the supply voltage is reduced to cut down leakage, a larger number of defects are sensitized, resulting in more failing cells within a memory array. This establishes a tradeoff between leakage reduction and yield of SRAMs.

Details of this part of our work are described in Chapter 3 of this thesis. The following submitted paper reports the results of this study:

1. A. Nourivand, A. J. Al-Khalili, and Y. Savaria, "Analysis of resistive open defects in drowsy SRAM cells," submitted to the *Journal of Electronic Testing: Theory and Applications (JETTA)*.

1.2.2 Modeling the Yield-Leakage Tradeoff in Large SRAM Arrays Considering Extreme Failure Events

A main contribution of this thesis is the modeling of the yield-leakage tradeoff in SRAM arrays. This analysis is essential for a design-time determination of the supply voltage to be applied to an SRAM subject to a target yield and leakage budget. Unlike the existing models, we have considered the impact of rare failure events, due to the extreme process parameters variations, on the yield-leakage tradeoff. We have employed concepts from extreme value theory (EVT) to model the rare failure events in SRAMs at scaled supply voltages. The results show that even a moderate scaling of the standby supply voltage results in significant yield losses in large non-fault-tolerant SRAMs, due to the failure of cells with extremely skewed process parameters. The yield losses grow with the size of memory and the aggravating process parameters variations.

The modeling methodology and the results of the yield-leakage tradeoff analysis are described in Chapter 4 of this thesis. The following paper summarizes the results of this part of our work:

2. A. Nourivand, A. J. Al-Khalili, and Y. Savaria, "Aggressive Leakage Reduction of SRAMs Using Fault-Tolerance Techniques: The Yield-Power Tradeoff," submitted to the *IEEE Transactions on Circuits and Systems I.*

1.2.3 Aggressive Leakage Reduction of SRAMs Using Fault-Tolerance Techniques

We investigated the aggressive leakage reduction of SRAMs using fault-tolerance techniques. Using the proposed model for yield-leakage tradeoff, it was shown that employing fault-tolerance techniques allows for efficient leakage reduction of SRAMs by providing tolerance to data-retention failures during the sleep mode. The results showed that repairing a memory by adding a small number of redundant resources or incorporating simple error correcting codes (ECC) allows for significant leakage reductions while incurring negligible yield losses. The combination of redundancy and ECC, however, allowed us to reach the bounds of the leakage reduction. In particular, the latter approach was shown to be viable when variations are large and the activity factor of memory is small.

The details of this investigation are reported in Chapter 4 of this thesis. The results of this part of our work are reported in the above-mentioned paper (i.e., paper No. 2).

1.2.4 Post-Silicon Tuning of Standby Supply Voltage for Reduction of Parametric Yield Losses Due to Data-retention Failures

We proposed a post-silicon standby supply voltage tuning technique for SRAMs to compensate for the die-to-die process parameters variations, and thereby decrease yield losses due to the parametric data-retention failures during the sleep mode. It was shown that applying an identical standby voltage to all dies, regardless of their specific process parameters variations, results in the failure of some dies, due to the data-retention failures, and thus it entails significant yield losses. To avoid yield losses,

we proposed to tune the standby voltage of each individual die to its corresponding minimum level. A test algorithm was presented to identify the minimum applicable standby voltage to each individual memory die after manufacturing. The effects of adding redundant resources on the minimum applicable standby voltage to a memory die was also investigated. Simulation results showed that yield can be enhanced significantly by the combined effect of repairing and standby voltage tuning, even when heavy process variations are present.

The details of this study are elaborated in Chapter 5 of this thesis. The following paper is submitted based on the results from this part of our research:

3. A. Nourivand, A. J. Al-Khalili, and Y. Savaria, "Post-silicon tuning of standby supply voltage in SRAMs to reduce yield losses due to parametric data-retention failures," accepted for publication in the *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*.

1.3 Organization of the Dissertation

The organization of this dissertation is as follows:

In Chapter 2, we provide the background on SRAM organization and operation, and we discuss the challenges of SRAM design in nanoscale technologies. The design paradox of low power and high yield is explained in this chapter.

In Chapter 3, results of fault injection and simulation of drowsy SRAMs are presented, and the newly observed single-cell static and dynamic drowsy faults are described. Then, a March test for detection of all drowsy faults as well as the simple traditional faults is proposed.

Chapter 4 investigates the aggressive leakage reduction of SRAMs using different

fault-tolerance techniques. A simulation methodology is presented for modeling the tail distribution of cell failures at scaled voltages. Then, mathematical relations are developed to compute the yield of a complete memory array from the failure probability of a single cell at scaled rail-to-rail voltages. Finally, the simulation results for a 64KB memory are presented, and the effectiveness of various fault-tolerance techniques for leakage reduction of SRAMs with minimal yield loss is analyzed.

In Chapter 5, a post-silicon standby supply voltage tuning scheme for SRAMs is presented to decrease yield losses due to the parametric data-retention failures during the standby mode, while reducing the leakage currents effectively. An implementation of the proposed tuning technique is demonstrated. The simulation results for the inter-die distribution of minimum applicable standby voltage of memory dies, and the corresponding yield enhancements by the proposed technique are presented.

Chapter 6 summarizes the contributions of this research and draws the main conclusions.

Chapter 2

Background

In order to study the design for the low power and high yield dilemma in SRAMs, an understanding of their organization and operation is required. Hence, in this chapter, we first provide a brief description of SRAMs architecture and operation. Then, the sources of power dissipation in SRAMs are discussed and the existing leakage reduction techniques are reviewed. The impact of these techniques on the stability of SRAM cells and the corresponding yield losses are evaluated.

2.1 SRAMs Organization and Operation

In the following, we briefly describe the organization and operation of SRAMs.

2.1.1 SRAMs Organization

An SRAM consists of an array of memory cells along with peripheral circuits that enable reading from and writing into the array. The basic organization of an SRAM array is shown in Figure 2.1. The memory array consists of 2^n rows and 2^m columns of cells. During a memory access, the supplied address is decoded by the row decoder

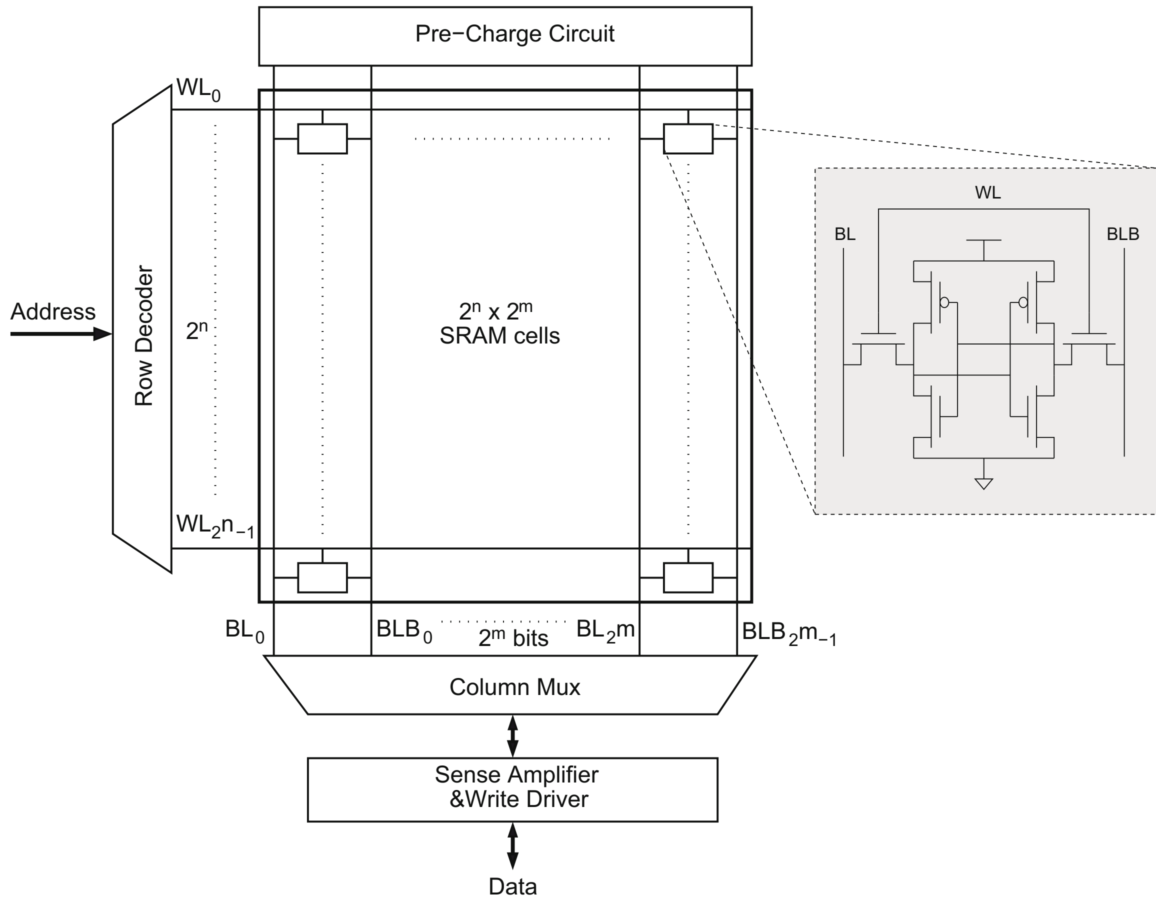


Figure 2.1. A typical SRAM organization.

to select one of the rows by activating its corresponding wordline (WL). To obtain a proper aspect ratio (length:width), multiple data words are usually placed in one row [12]. Thus, a column multiplexer (MUX) is used to select only the target data word. Data words are usually a group of 16, 32, or 64 bits. For example, in a memory with a 32-bit data width and $2^m = 256$ bits per row, each row contains 8 data words.

During a read operation, the bitlines (BL and BLB) are first charged to VDD by the pre-charge circuit (see Figure 2.1). Then, the wordline of the accessed row is activated and the BLs (BLBs) starts to discharge if their corresponding cell contains data '0' ('1'). Sense amplifiers are used to detect a very low differential voltage between BL and BLB of each column, speeding up the read cycle. A timing control

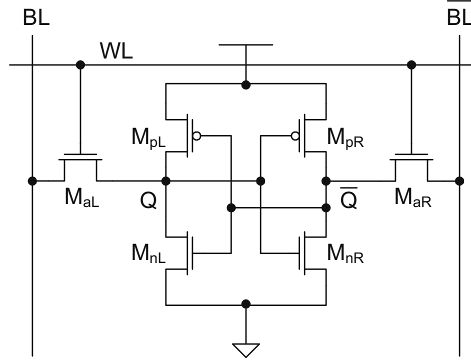


Figure 2.2. Conventional 6T SRAM cell.

unit usually triggers the sense amplifiers at the right time, so they capture and send the correct data to the I/O drivers. For a write operation, the write drivers charge or discharge the bitlines according to the input data and then the corresponding wordline is activated to write the data into the cells.

2.1.2 SRAM Cells

SRAM cells have a latch type structure that enables them to retain their data as long as the power supply is maintained. Different SRAM cells have been proposed in the literature, e.g., 4-transistor (4T) [12], 6T [12], 8T [13, 14], 10T cells [15]. However, the 6T cell is still the most popular option for embedded memory design at the present time due to its small area and stable operation.

A schematic of the conventional 6T SRAM cell is shown in Figure 2.2. The two pull-down transistors (MnR, MnL) and the two pull-up transistors (MpR, MpL) comprise a pair of cross-coupled inverters which operates as a static latch to store one bit of data [12]. Access to the storage nodes, T and F, for reading and writing is enabled by wordline WL which controls the two access transistors MaR and MaL.

The two bitlines, BL and BLB, transfer both the stored data and its inverse in and out of the cell.

The size of the cell should be as small as possible to achieve high memory density and high yield. However, reliable operation of the cell imposes some sizing constraints. In particular, a careful sizing of the transistors is necessary to avoid a destructive read. The read operation can be destructive because the access and the pull-down transistors are in conflict during the read time and the voltage of the low storage node rises to a voltage higher than ground. The cell ratio, defined as $r = W_{pull-down}/W_{access}$, controls the voltage rise and it must be large enough to prevent the voltage of node '0' from rising above the driver transistor's threshold voltage [12]. A reliable write operation, on the other hand, is ensured if the access transistor can overcome the pull-up transistor and pull down the voltage of storage node '1' to a voltage lower than the threshold voltage of the pull-down transistor. The cell pull-up ratio, defined as $q = W_{pull-up}/W_{access}$, must be small enough to ensure that the storage node voltage is pulled below the driver transistor threshold voltage, allowing the cell to flip [12].

2.2 Leakage Power Dissipation in SRAMs

Power is dissipated as leakage and active switching in SRAMs. Due to a low switching factor in SRAMs, leakage power tends to be the dominant part of the power consumption [1, 6]. The subthreshold leakage current, gate-tunneling current, and the reverse-biased junction current are known to be the major components of the leakage consumption in sub-100nm technologies [16]. The results in this work are based on simulations using an industrial 90-nm technology and a predictive 45-nm technology (PTM) [17]. Our simulations show that the junction current is negligible compared to the other two leakage mechanisms in these technologies. Therefore, we

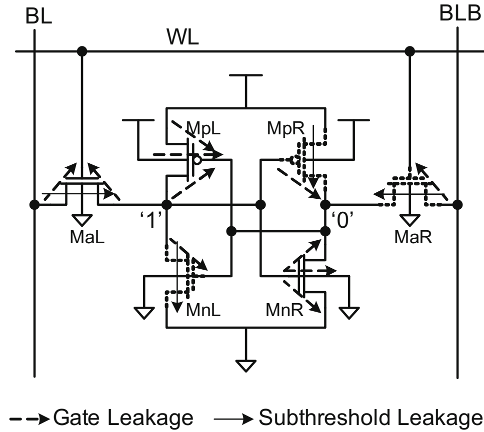


Figure 2.3. SRAM cell leakage currents during standby mode. The leaking transistors are shown in dotted lines.

will consider only the subthreshold and gate leakage currents in this work. Figure 2.3 shows the main leakage contributors in a 6T SRAM cell.

2.2.1 Subthreshold Current

The subthreshold current in MOSFETs is the off-state leakage current from drain to source of the device. As supply voltage scales down with technology, the transistor threshold voltage is scaled down as well in order to maintain performance. Due to the exponential dependence of subthreshold leakage current on the threshold voltage [12], it is exponentially growing with technology scaling [6].

2.2.2 Gate-Tunneling Current

The gate-oxide thickness, t_{ox}, is rapidly decreasing with each technology node to achieve higher speeds [6]. A thin gate-oxide layer of less than 2-3 nm can cause a dramatic increase in gate-tunneling currents. However, the introduction of new gate-dielectric materials with high dielectric constant (high-k) beyond 45nm technologies has reduced the gate-tunneling currents significantly [18].

2.3 SRAM Leakage Reduction Techniques

SRAM leakage reduction techniques can be broadly categorized into state-preserving and non-state preserving. State-preserving techniques do not alter the contents of the memory, while in non-state-preserving techniques the data is lost. Non-state-preserving techniques generally save more leakage by completely removing the power from SRAMs. The latter techniques are only applicable if a copy of the data is retained in some other place, e.g., a higher level memory. Therefore, they can be applied to write-through caches, for example. However, shutting off the memory can incur a significant dynamic power overhead due to the induced misses that require accesses to higher level memories. Therefore, state-preserving techniques are preferred for caches despite their lower leakage reduction capabilities [19, 20]. These techniques have been widely applied to instruction and data caches at all hierarchy levels, i.e., L1, L2, and L3 [19, 20, 21, 22].

Various techniques, e.g., voltage scaling [19], source biasing [23], and body biasing [24] have been proposed in the literature to reduce leakage power of SRAMs by switching the cells into a state-preserving low-leakage mode during the idle periods [25, 26]. Scaling the rail-to-rail voltage of SRAMs, by voltage scaling or source biasing, is a more attractive technique due to its lower cost and higher leakage savings [27]. Thus, in this work, we focus on the leakage reduction of SRAMs using the dynamic voltage scaling and source biasing techniques.

2.3.1 Supply Voltage Scaling

Figure 2.4(a) shows the supply voltage scaling technique for a typical SRAM cell. When the cell is in the active mode, i.e., $Sleep = 0$, the nominal supply voltage (V_{DD}) is applied to the cell. However, when cells are idle, they are placed in a *drowsy mode*

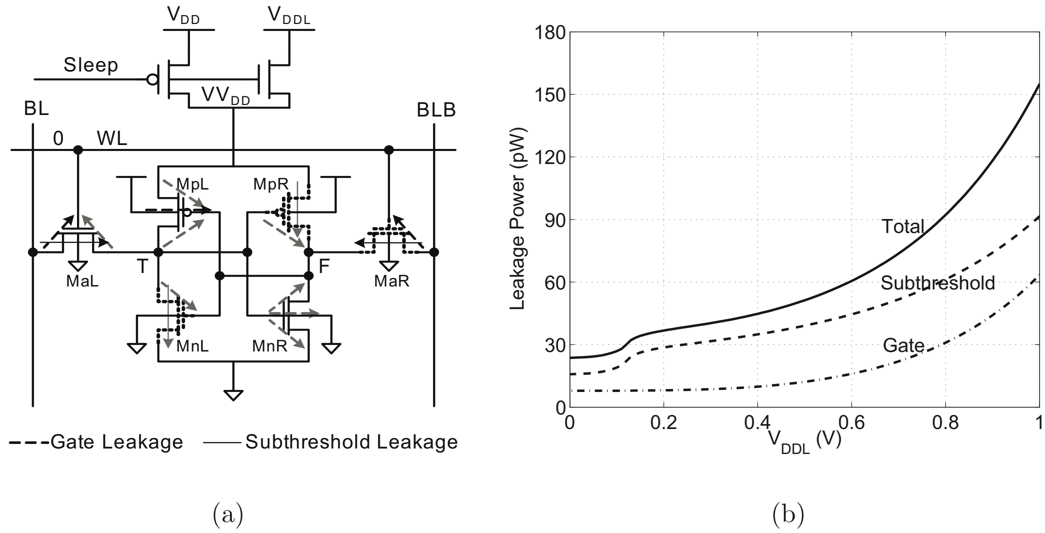


Figure 2.4. (a) SRAM cell leakage currents at reduced supply voltage and (b) circuit simulation results for leakage currents at reduced supply voltages for an SRAM cell in a 45-nm technology at $T = 27°C$.

by activating *Sleep* signal, where a low standby voltage (V_{DDL}) is applied to the cell. The leakage power is significantly reduced in the drowsy mode due to the decreases in both subthreshold and gate leakage currents. The reduced leakage currents are shown in gray in Figure 2.4(a). The leakage current versus supply voltage of a typical SRAM cell in a 45-nm technology is shown in Figure 2.4(b). As can be seen, the leakage currents reduce sub-linearly with the standby voltage.

Scaling of the supply voltage during the drowsy mode can be alternatively realized by power gating the cells using a large sleep transistor as shown in Figure 2.5 [18]. This removes the need for an additional on-chip supply voltage, i.e., V_{DDL} During the drowsy mode, the large sleep transistor is turned off, and thus the virtual VDD node starts to discharge due to the SRAM leakage currents. To stabilize the voltage of virtual VDD node at a pre-defined level (V_{DDL}), small bias transistor(s) are placed in parallel with the sleep transistor (see Figure 2.5). To allow for post-silicon compensation of process variations impact, the bias transistors are made programmable, e.g., through fuses [18].

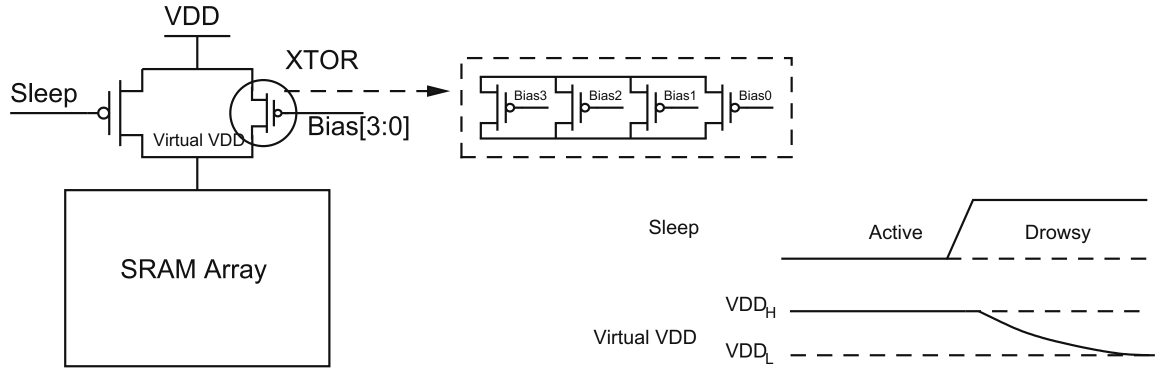


Figure 2.5. SRAM supply scaling by power gating. The programmable bias transistors enable controlling of the virtual VDD.

2.3.2 Source Biasing

The rail-to-rail voltage of SRAM cells can be alternatively scaled by raising the voltage of their source line. Figure 2.6(a) shows a typical SRAM cell with the source-biasing technique. During the active mode the */Sleep* signal is high and thus the virtual GND node is tied to ground. To switch the SRAM to the drowsy mode, the */Sleep* signal is set low and a higher supply voltage (V_{SB}) is applied to the source-line of cells. Both the subthreshold and gate leakage currents are affected by source-biasing. The reduced leakage currents are shown in gray in Figure 2.6(a). The leakage current versus source biasing voltage (V_{SB}) of a typical SRAM cell in a 45-nm technology is shown in Figure 2.6(b). As can be seen, the leakage currents reduce efficiently with the raising of the source-bias voltage.

Similar to the power gating, the SRAM cells can be ground gated to raise the virtual ground voltage of cells. Figure 2.7 shows the ground gating technique, where a sleep transistor is used to cut off the ground node of an SRAM array during the sleep mode. As a result, the virtual ground node is charged to a predefined level (V_{SB}) by the leakage currents of the SRAMs, eliminating the need for an extra supply voltage. The bias transistors are used to tune the upper limit of the source-bias voltage. To

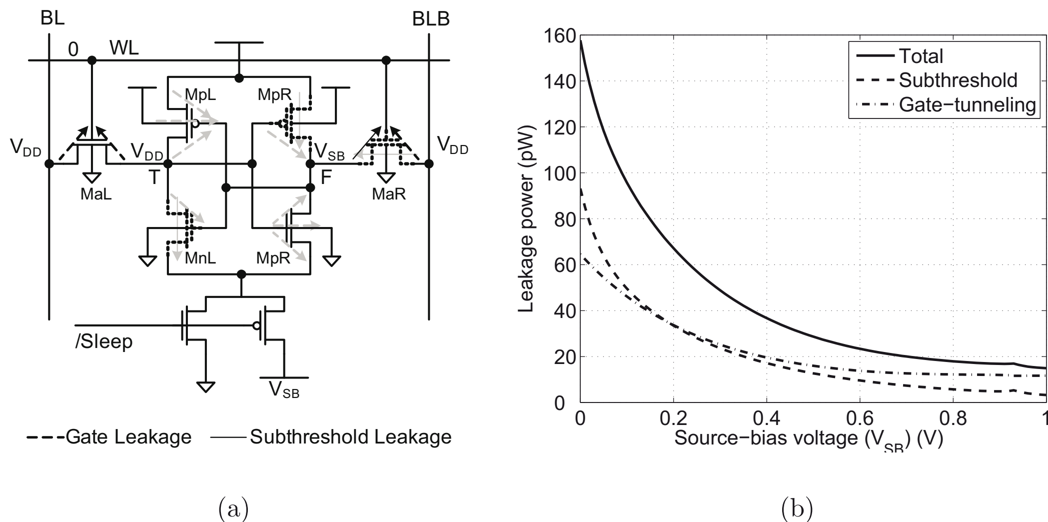


Figure 2.6. (a) SRAM cell leakage currents at raised source-line voltage and (b) circuit simulation results for leakage currents at raised source-line voltages for an SRAM cell in a 45-nm technology at $T = 27°C$.

compensate for the process variations effect on the virtual ground node voltage, bias transistors are made programmable [18, 28].

2.3.3 Architectural Level Leakage Reduction Techniques

Architectural level leakage reduction techniques work together with the circuit level techniques, presented in the previous section, to reduce the leakage power dissipation. Voltage scaling and source-biasing techniques are equally applicable to SRAM cells in all memory structures.

In general, cache leakage reduction techniques can be divided into two categories [27]: i) passive leakage reduction and ii) active leakage reduction techniques. In passive leakage reduction techniques, the whole memory is switched to the sleep mode during the idle periods of the system. Whereas, in active leakage reductions, only portions of the memory are dynamically switched between active and drowsy (sleep) modes during the system run-time. At any time window, the accesses to memory are usually concentrated only on a small subset of the active lines, and all the other lines

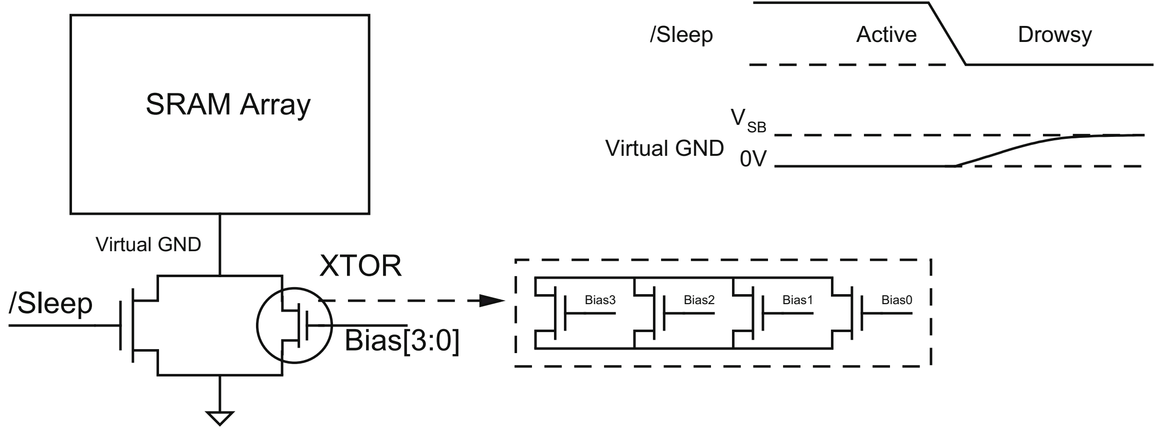


Figure 2.7. SRAM ground gating. The programmable bias transistors enable to control the virtual GND.

are idle, dissipating leakage power. Thus, active leakage reduction techniques achieve a higher leakage reduction efficiency compared to the static techniques [29].

In active leakage reduction techniques, blocks of memory, at different granularity, are dynamically activated and deactivated based on a mode management policy. Coarse-grained techniques reduce the hardware overhead by employing policies that apply to large blocks of cache, while fine-grained techniques suppress leakage at small blocks of cache at the cost of extra overhead. In order to obtain the best power saving results with the minimal performance penalty, the access profile of a memory structure needs to be considered when determining the following parameters:

- Sleep granularity: the size of the smallest block of cells which can be switched to the sleep mode independently, e.g., row-by-row, bank-by-bank.
- Mode management policy: the policy that manages the switching of memory blocks between the active and sleep operating modes.

The cache management policies can be categorized as: 1) per-access wake-up, 2) periodic sleep, and 3) wake-up counter. The above techniques are described below.

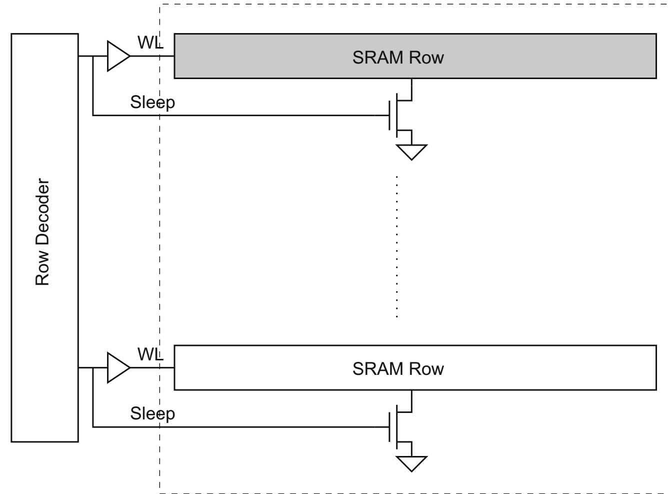


Figure 2.8. DRG-Cache.

2.3.3.1 Per-access wake-up

With this policy, only the row (bank) that is going to be accessed is awakened, and then it is put back in the sleep mode immediately after the access. This policy is implemented in the data-retention gated-ground cache (DRG-Cache) [21] as shown in Figure 2.8. In this scheme, all the cells in a row share a common sleep transistor that is activated by the row's wordline. Hence, the cells are turned on only during the access times. Another leakage reduction architecture called segmented virtual grounding (SVGND) [30] implements this policy by ground gating columns of cells. This policy is the simplest, however it can incur large power overheads due to the frequent switchings between active and sleep modes. For L2 caches, a row-by-row drowsy scheme using the per-access wake-up policy is proposed in [21]. However, power saving is reported to be only about 50% due to the large dynamic power overhead of frequent switching between active and drowsy modes.

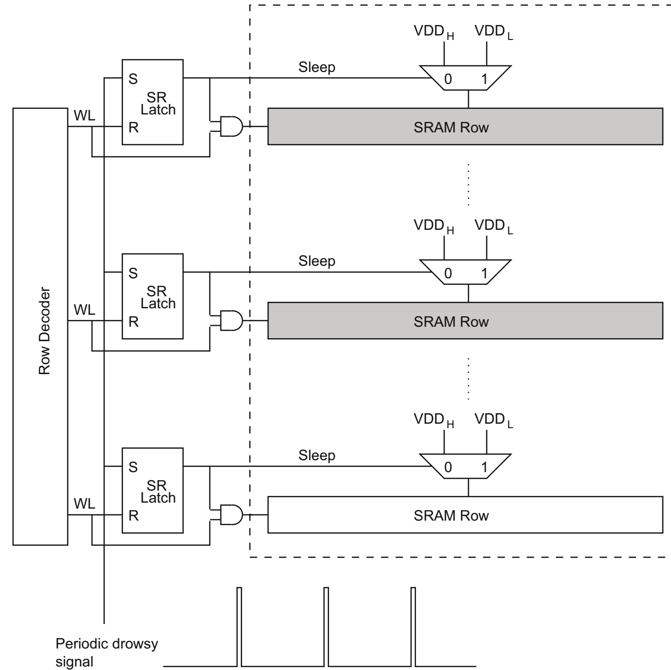


Figure 2.9. Leakage reduction technique using periodic sleep policy.

2.3.3.2 Periodic sleep

Due to the temporal locality of reference in caches, an accessed data line will most probably be accessed again after a short period of time. Thus, it is more efficient to keep an accessed line alive for a period of time after its first access. Periodic sleep exploits this property in caches to reduce the power overhead by removing the unnecessary switchings between the active and sleep modes. In this policy, when a line (bank) is about to be accessed it is awakened and is kept in the active mode. However, the whole memory is periodically put into the sleep mode by a global periodic signal. The period of this signal is determined so that the optimum energy efficiency is obtained [19]. This policy is implemented for a data cache in [19], and supply voltage scaling is employed as the leakage reduction technique. The architecture of this technique is shown in Figure 2.9. To store the state of each row, an SR latch is used, which is reset when a row is accessed for the first time, applying VDD_H to the row. All the latches are set every 2000 clock cycles, to switch the whole memory array to

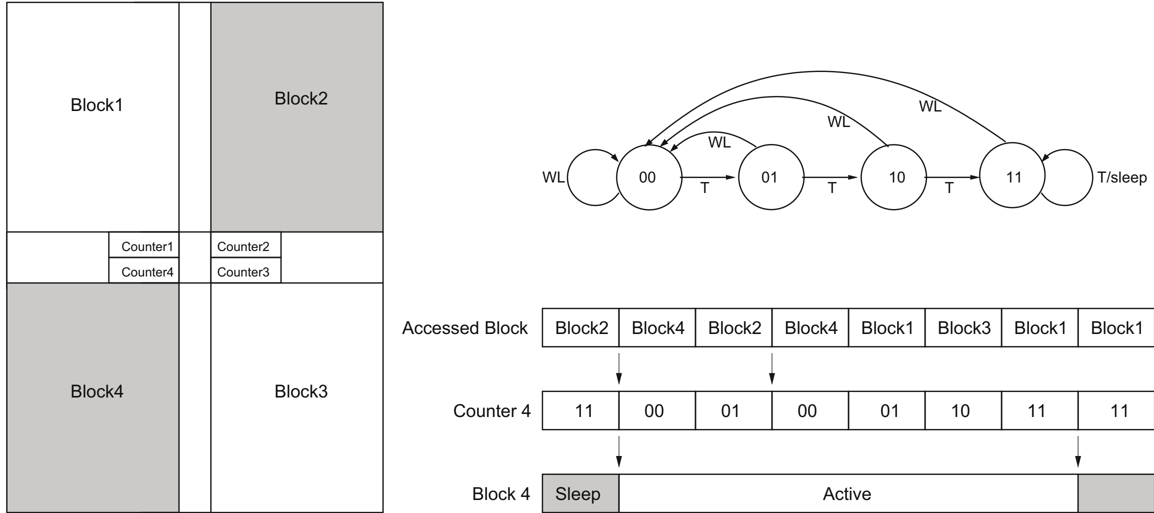


Figure 2.10. Leakage reduction with wake-up counters.

the drowsy mode, by applying a lower supply voltage (VDD_L). For L1 data caches, authors in [19] reported that a fine-grained, i.e. row-by-row, drowsy scheme with periodic sleep policy, achieves $60\% - 75\%$ leakage power reduction across SPEC2000 benchmarks [19].

2.3.3.3 Wake-up counter

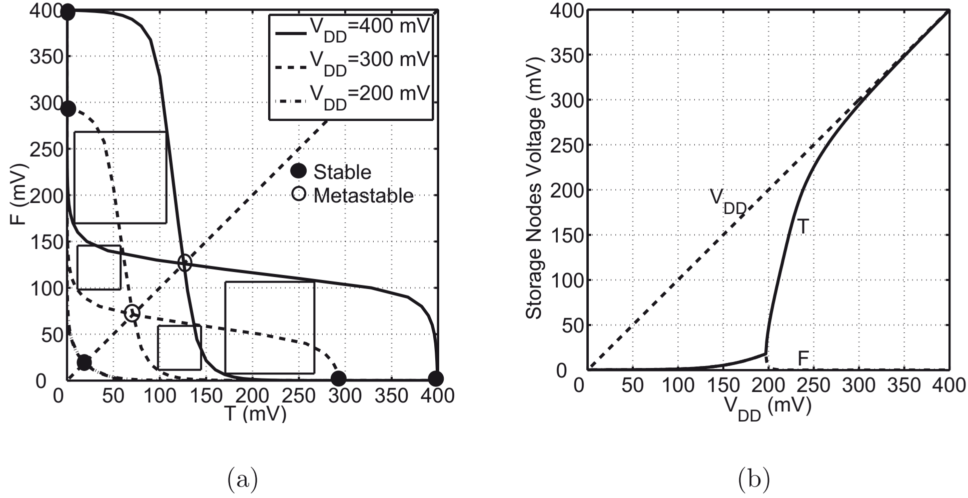
Another technique to reduce the power overhead due to unnecessary switching between active/sleep modes is to use a wake-up counter. Here, a counter is associated with each bank (row) of the memory that switches it to the sleep mode if a certain time interval elapses from its last access. This interval is determined as the break-even point between leakage power and switching power overhead. Figure 2.10 shows the operation of this technique on a block-based memory. The wake-up interval is assumed to be 4 clock cycles in this example. The counter associated with a bank is reset if an access is issued to a data in that bank, otherwise, the counter is incremented. When counter counts up to 4, it activates the sleep signal of the bank, placing it in the sleep

mode. The counter and the operating mode of $Block4$ are shown in Figure 2.10 for a sample access scenario. At the first access to $Block4$, its counter is reset and thus it is awakened. At the second access to $Block4$, the counter is reset and thus the block is still kept in the active mode. Eventually, as there is no access to $Block4$ for the next 4 cycles, it is automatically placed in the sleep mode. The sub-array based drowsy scheme proposed for L3 caches [22] reported as about 95% leakage reduction using 16 clock cycle wake-up intervals.

2.4 Yield Losses Due to the Introduction of a Drowsy Mode to SRAMs

To maximize the leakage reductions in SRAMs, it is desirable to reduce the rail-to-rail voltage of cells as low as possible during the standby mode [31]. However, this also causes the SRAM cells to become less stable and thus fail to retain their data reliably. Indeed, an SRAM cell is capable of retaining data as long as its static noise margin (SNM) is positive. SNM of an SRAM cell is an accepted measure of the stability, and is defined as the minimum dc noise voltage necessary to flip the state of a cell. A graphical representation of SNM is presented by drawing the transfer characteristic of a cell's left inverter and the mirror transfer characteristic of its right inverter and finding the side of the maximum square nested between these two curves [32] as shown in Figure 2.11(a).

Switching memory cells to a drowsy mode, e.g., by lowering the supply voltage, reduces their noise margin [33, 19], as shown in Figure 2.11(a). As can be seen, as long as the SNM is larger than zero, the cell has two stable states, thus it retains its data. At $V_{DD} = 200mV$, SNM becomes zero and the regenerative effect of cross-coupled inverters of an SRAM cell is disabled. At this supply voltage, the voltage of both



(a) (b)

Figure 2.11. (a) Butterfly curve of a balanced cell at different supply voltages. SNM of the cell is reduced to zero at $V_{DD} = 200mV$, and (b) Waveforms for the voltage of storage nodes, i.e., T and F, of a balanced cell as the supply voltage is reduced down to zero.

storage nodes converge to a common stable point. The behavior of the storage nodes of a perfectly balanced cell as the supply voltage is reduced down to zero is shown in Figure 2.11(b), assuming that the initial state of the cell is '1', i.e., $T = 1.2V$ and $F = 0V$ (see Figure 2.2). As V_{DD} is reduced, the true node (T) follows it and the false node (F) remains at zero. However, below $300mV$, T starts to deviate from V_{DD}, and eventually at $V_{DD} = 200mV$, both nodes converge to a certain voltage and the state of the cell is lost. The supply voltage at which the SNM of a cell shrinks to zero is called its data-retention voltage (DRV).

Excessive process parameters variations and manufacturing imperfections in nanoscale technologies are increasingly resulting in "weak cells" with a severely degraded stability [34, 35, 36]. Switching cells to a drowsy mode reduces the cells stability, however, weak cells, which are inherently less stable, can be severely affected. Introduction of a drowsy mode to SRAMs can result in failure of the weak cells, and thereby degrade yield drastically. In the following, we explain the impact of these two factors on the failure probability of SRAM cells.

2.4.1 Impact of Process Variations on Drowsy SRAMs

As process geometries continue to shrink, controlling the variation in device parameters during fabrication is becoming increasingly difficult [10, 7]. The variations in device features can be either due to systematic or random variations in the fabrication process. Systematic variations are classified as across-field and layout-dependent variations [37]. Across-field systematic variations are caused by lithographic and etching sources such as dose, focus and exposure variations etc. [37]. These variations exhibit a strong spatial correlation and thus cause discrepancies in the behavior of identical devices at different locations on a photo-mask reticle. The layout-dependent systematic variations, on the other hand, can cause two layouts of the same device to have different characteristics even when they are located close to each other. Systematic variations are predictable and can be modeled based on factors such as layout structure and the surrounding topological environment [37]. Random variations, on the other hand, are unpredictable and are caused by random uncertainties in the fabrication process such as microscopic fluctuations in the number and location of dopant atoms in the channel region, gate line-edge and line-width roughness, (LER) and (LWR) respectively, [37]. Random variations can cause significant mismatch among two identical devices placed next to each other. These random variations are intrinsic to devices as they cannot be eliminated by external control of manufacturing processes or layout techniques [37].

Depending on the scale of variations, they are classified as inter-die (die-to-die) and intra-die (within-die) variations. Inter-die variations are caused due to systematic variations from lot-to-lot, wafer-to-wafer, and within-wafer variations, and affect every element on a chip equally (see Figure 2.12). However, intra-die variations are caused by both the systematic and random variations and result in discrepancies among properties of identical devices on the same chip as shown in Figure 2.12.

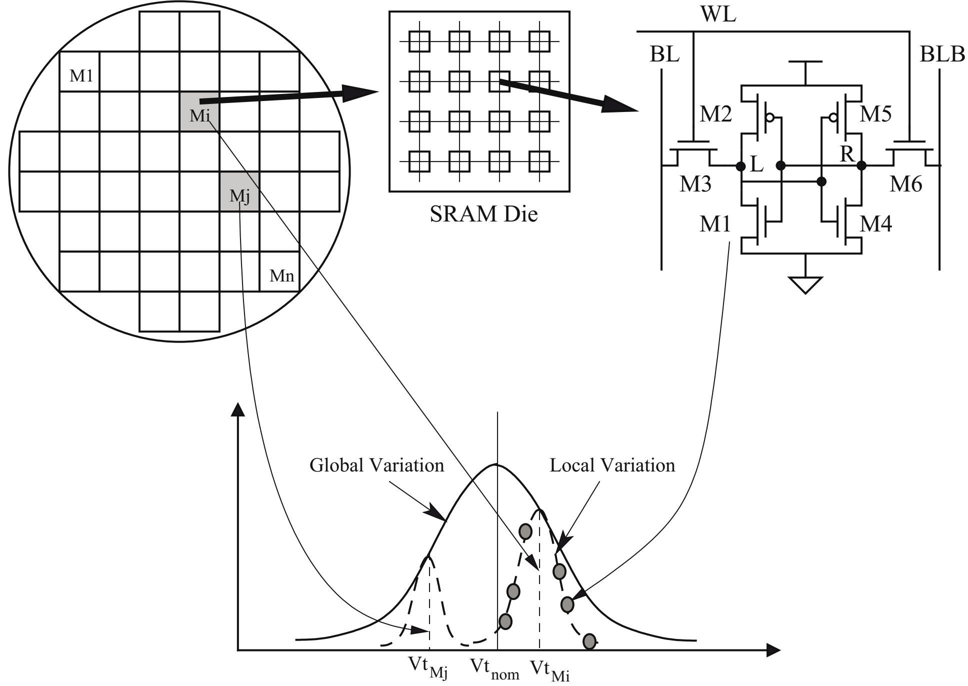


Figure 2.12. Inter and intra die variation modeling.

Traditionally, the inter-die fluctuations have been the main concern in CMOS digital circuit designs, and the intra-die fluctuations have been neglected [38]. However, in new technologies the intra-die variations have exceeded the inter-die fluctuations [38]. The intra-die variations contain both the systematic and the random components of the process parameters variations. Random variations are more concerning as the systematic variations can be minimized by layout techniques [37]. The random variations effect the current drive capability of transistors by causing variations in the threshold voltage and channel dimensions of the device. Threshold voltage variation due to RDFs in the channel area is the most dominant source of variation in current technologies [7]. Variation in the threshold voltage (Vt) of transistors is inversely proportional to the square root of the channel area [39] (see Figure 2.13). With technology scaling, the random variations are becoming the dominant part of intra-die variations. For example, σVt as large as 45mV is reported for the Intel's 45-nm technology as shown in Figure 2.13. As minimum-size transistors are used in

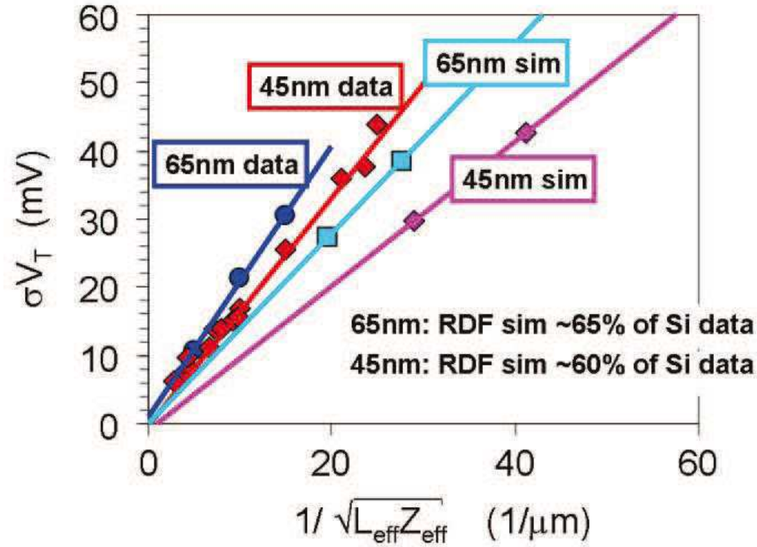
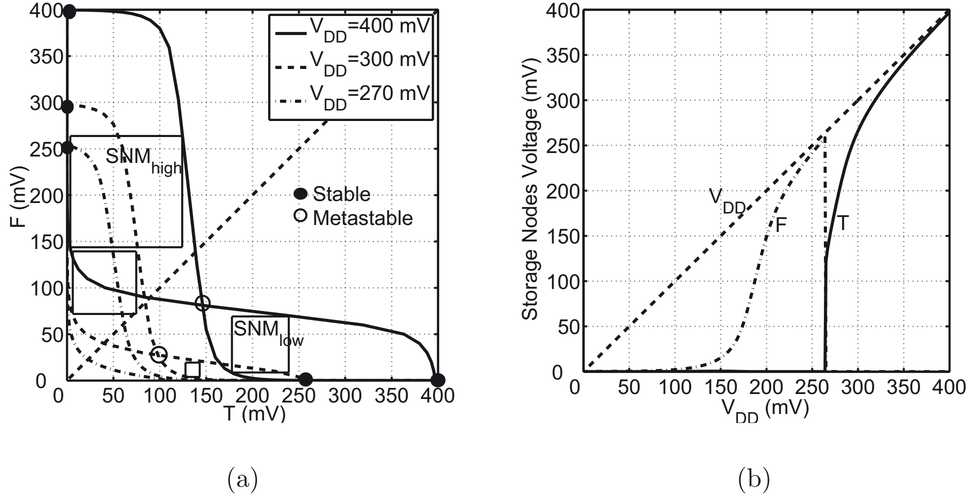


Figure 2.13. Random within-die variations in threshold voltage in 65-nm and 45-nm technologies. Adapted from [7] (Copyright 2008 Intel).

SRAMs to obtain a high density, the random variations are accentuated causing significant fluctuations in the cells performance, stability, and leakage. These variations pose a growing threat to the yield of memory chips [40].

Process parameters variations impact the data-retention capability of SRAM cells by shifting the parameters of its transistors. In particular, mismatches among transistors of a cell result in an imbalanced cell with a much weaker data-retention capacity. The butterfly curves of an asymmetric cell are shown in Figure 2.14(a) as an example. As can be seen, the SNM on the left and right lobes of the curve are different. Hence, the cell's SNM is defined as their minimum. In this cell, as V_{DD} is reduced, SNM_{low} decreases to zero before SNM_{high} (see Figure 2.14(a)). If V_{DD} is reduced beyond this point, the cell flips to its more stable state, i.e., '0'. Therefore, the DRV of an asymmetric cell is defined as the V_{DD} at which the minimum of SNM_{high} and SNM_{low} becomes zero. Figure 2.14(b) shows the behavior of the storage nodes of this asymmetrical cell as the supply voltage is reduced down to zero, assuming that the initial state of the cell is '1', i.e., $T = 1.2V$ and $F = 0V$. As V_{DD} is reduced, the



(a) (b)

Figure 2.14. (a) Butterfly curves of an imbalanced cell at different supply voltages. SNM_{low} is reduced to zero before SNM_{high} at $V_{DD} = 270mV$. and (b) waveforms for the voltage of storage nodes, i.e., T and F, of an imbalanced cell as the supply voltage is reduced down to zero.

true node (T) follows it and the false node (F) remains at zero. However, when the supply voltage falls below $270mV$, the cell flips to state '0'. Thus, the DRV of this imbalanced cell is higher than that of a balanced cell.

In practice, inter-die (die-to-die) and intra-die (within-die) variations in process parameters result in a statistical distribution of DRV of SRAM cells [41, 42, 43]. For example, the histogram of DRV obtained by 5000 Monte Carlo (MC) simulations in a 45-nm technology is shown in Figure 2.15. The maximum applicable source-bias voltage to such a memory array is determined by the cell which has the smallest V_{SBmax}. Therefore, the upper bound of DRV from this histogram needs to be determined as the minimum VDD_L that can be applied to this memory [44, 45].

2.4.2 Impact of Defects on Drowsy SRAMs

Most faults in memory circuits are caused by spot defects (SD). SDs can be modeled as spots of extra, missing or undesired material, and can cause undesired

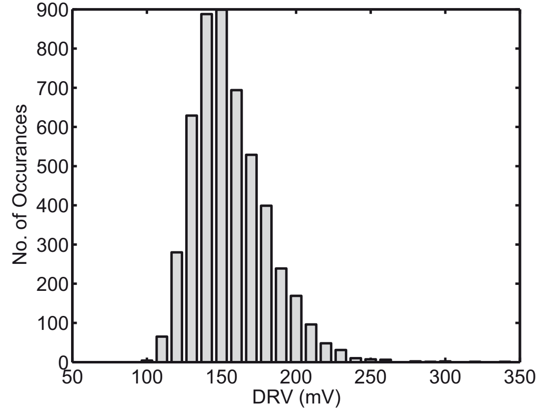
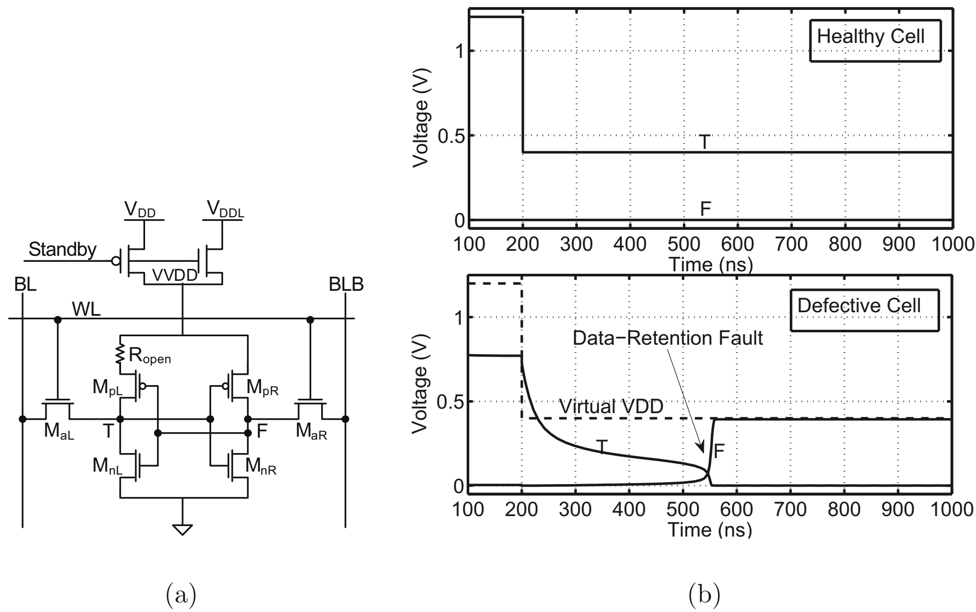


Figure 2.15. Histogram of the DRV from a 5000 point Monte Carlo simulation of SRAM cells in a 45-nm predictive technology node.

shorts or opens in circuits [46]. Manufacturing defects can significantly impact the data-retention capability of SRAM cells. Strong opens, i.e., $R_{op} \rightarrow \infty$, or strong shorts, i.e., $R_{sh} \approx 0$, usually cause an SRAM cell to malfunction during normal operating conditions, and thus they are detected by March algorithms performed at the active operating mode. However, defects can still connect nodes weakly by having a finite parasitic resistance, causing weak opens or weak shorts/bridges. Such weak defects let the SRAM cells still function, although poorly. However, they can turn into a strong fault at deteriorated operating conditions, e.g., reduced supply voltage during the drowsy mode.

For example, a defective SRAM cell with a resistive open defect in the pull-up path, as shown in Figure 2.16(a), will fail to retain its data at higher standby voltages compared to a healthy cell, due to smaller currents from the pull-up path. Hspice simulation results for the voltages of storage nodes, i.e., T and F, of a healthy cell and such a defective cell, assuming $R_{open} = 30M\Omega$, are shown in Figure 2.16(b). As can be seen, a healthy cell is capable of retaining data during the drowsy mode, when the supply voltage is reduced down to $0.4V$. However, the defective cell loses its data some time after being placed in drowsy mode, resulting in a data-retention fault.



(a) (b)

Figure 2.16. (a) A resistive open defect in pull-up path of a 6T SRAM cell, (b) simulation results showing the reduction of DRV in defective SRAM cells.

2.4.3 Importance of Extremal Events in Large SRAMs

SRAMs are a yield limiter in modern integrated circuits due to their large silicon area. Modern processors and SoCs contain large embedded SRAM blocks with millions of replicated bitcells. Even an extremely small failure probability of cells can be magnified by the large number of SRAM cells. A memory array without a repair mechanism will be rendered nonfunctional if it contains even a single failing cell. Thus, even very small cell failure probabilities can translate to significant yield losses in large SRAM arrays. For example, for a 1Mb memory array, a cell failure probability as low as 10^{-8} can result in 1% yield loss. Cell failures can occur due to either the manufacturing defects or the excess process parameters variations. Hence, in large memory arrays, with millions of replicated bitcells, the rare variation events resulting in cells with a DRV extremely deviated from its typical value needs be taken into account.

In fact, a close examination of Figure 2.15 reveals that DRV distribution tends to

exhibit a heavy right tail, which is not fully represented due to the limited number of MC simulations. The rare events in the tail of this distribution might not be an issue in small memory arrays, as their probability of occurrence is extremely low. However, the probability of these rare events can be magnified in large memory arrays due to the large number of cells on a single chip [47, 48]. If the applied standby supply voltage (V_{DDL}) to a memory array exceeds DRV of these rare event cells, they will fail to retain their data during the sleep mode, resulting in the DRFs. These cell failures, in turn, can cause the whole array to fail, entailing significant yield losses. For example, with a small 64KB memory, even a cell failure probability as low as 1.9×10^{-8} will result in 1% yield loss.

2.4.4 Yield-Leakage Tradeoff in SRAMs

As mentioned earlier, to reduce leakage currents more efficiently, it is desirable to push the drowsiness level of cells as high as possible, by lowering the supply voltage. However, due to a long tail distribution of cells' DRV, the cell failure probability, and thereby array failure probability, can drastically increase as the cells' voltage is reduced, causing large yield degradations. As the standby voltage is reduced for more aggressive leakage reduction, larger yield losses will be entailed. Thus, there is a trade-off between leakage reduction and yield of SRAMs. Efficient techniques are, therefore, essential for a yield-aware leakage reduction of SRAMs.

2.5 Summary

In this chapter we briefly described the organization and operation of SRAMs. Then, two of the major challenges of SRAM design in nanometer era, namely excess leakage power dissipation and yield losses due to the process parameters variations

and manufacturing imperfections, are discussed. Different circuit and architectural level leakage reduction techniques for SRAM are reviewed. The impact of these techniques on the cells robustness are discussed. In particular, we showed that leakage reduction by voltage scaling can also result in diminished data-retention capability, and thus increase the probability of data-retention failures during the standby mode. In summary, the discussions presented in this chapter justify the need for a yield-aware leakage power reduction of SRAMs, for which we make an attempt to provide solutions in the following chapters.

Chapter 3

New Fault Models and Their Impact on Low Leakage Drowsy SRAMs

New fault behaviors can arise by introducing drowsy mode to SRAMs. These new faults may not be fully covered by test algorithms/techniques that are applied at normal operating conditions. Nevertheless, they can cause failure in the memory when it is switched to the drowsy mode. Therefore, it is imperative to test memories at all operating modes, in order to minimize the defects per million (DPM). In this chapter, we develop fault models for the erroneous behaviors that emerge when SRAMs are switched to a drowsy mode. Then, based on the derived models, a new March test is proposed that is capable of detecting all drowsy faults as well as the simple traditional faults.

3.1 Impact of Defects on Drowsy SRAMs

Most faults in memory circuits are caused by spot defects (SD). SDs can be modeled as spots of extra, missing or undesired material, and can cause undesired shorts or opens in circuits [46]. Depending on their conductivity in memory chips, SDs cause alterations that can be categorized into one of the following three groups:

1. Open: An undesired resistance R_{op} within a connection, where $0 < R_{op} < \infty$.
2. Short: An undesired resistive path between a node and V_{DD}/GND. The resistor value, called R_{sh}, is given by $0 \leq R_{sh} < \infty$.
3. Bridge: An undesired resistive path between two nodes other than V_{DD}/GND. The resistor value, called R_{br}, is given by $0 \leq R_{br} < \infty$.

Defects with a finite resistance are called resistive defects.

Strong opens, i.e., $R_{op} \to \infty$, or strong shorts and bridges, i.e., $R_{sh} \approx 0$ or $R_{br} \approx 0$, usually cause an SRAM cell to malfunction and thus, they are easily detected by March algorithms. However, defects can still connect nodes weakly by having a finite parasitic resistance, causing weak opens or weak shorts/bridges. Such weak defects let the SRAM cells still function, although poorly. Based on the severity of the symptoms of a defect, memory faults are categorized as [46]:

1. Strong fault (sF): A fault which is fully sensitized by an operation, i.e., a read/write operation fails.
2. Weak fault (wF): A fault which is partially sensitized by an operation; e.g., a defect that creates a small disturbance of the voltage of the true node (T) of the cell.

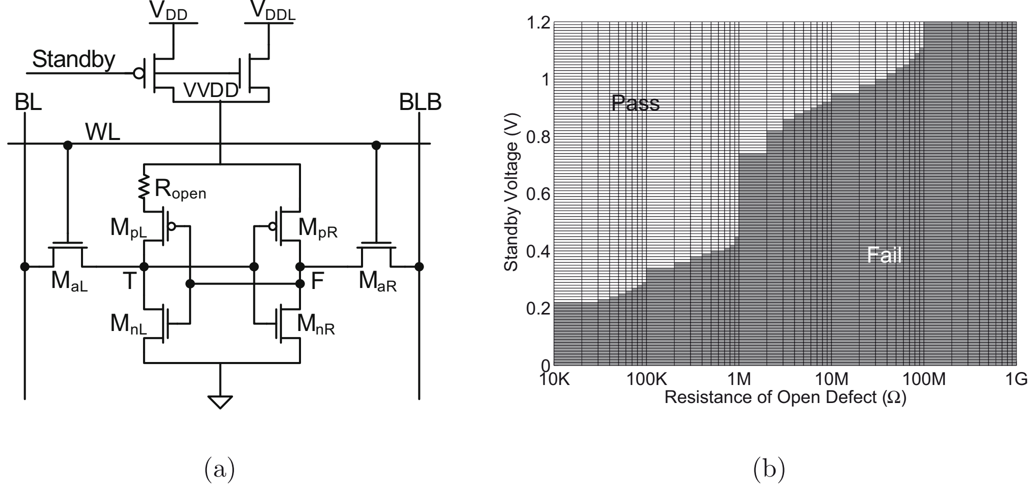
This means that in the presence of weak faults, all operations, i.e., read and write, pass correctly. However, a weak fault has the potential to turn into a strong fault at deteriorated operating conditions, e.g., reduced supply voltage during drowsy mode.

Defects can impact two important characteristics of SRAM cells which are detrimental for the fault-free operation as well as the performance of drowsy memories: (i) the minimum standby voltage that SRAM cells can tolerate without losing their data, so-called the data retention voltage (DRV), and ii) the minimum time required to transition from drowsy to active mode, called the wake-up time.

3.1.1 Data Retention Voltage (DRV) of Defective Cells

Leakage currents of SRAMs are reduced sub-linearly with the reduction of supply voltage. Hence, it is desirable to reduce the standby voltage down to the DRV of healthy cells to save as much leakage power as possible. However, the more the supply voltage is reduced, the weak faults are more sensitized, resulting in failures. This is because the minimum tolerable standby voltage of a weak cell depends on the location and the resistance value of its defect. Generally, the larger the parasitic resistance of an open defect is, the higher the cell's minimum standby voltage will be. This means that a cell with a large open defect fails at higher standby voltages compared to a cell with a small defect. In addition, a defect is sensitized at different standby voltages depending on its location in the cell. Therefore, the level of the standby voltage determines whether a defective cell will exhibit faulty behavior if it is switched to the drowsy mode.

To determine the influence of location and resistance value of defects on the minimum standby voltage of SRAM, we did fault injection and simulation on a typical SRAM cell for an open defect in pull-up path of the cells as shown in Figure 3.1(a). Each open defect is modeled by a single resistance ranging in value from 100Ω to $1G\Omega$



(a) (b)

Figure 3.1. (a) A resistive open defect in pull-up path of a 6T SRAM cell, (b) Shmoo plot showing pass/fail status of an SRAM cell for two parameters: i) resistance of open defect and ii) standby voltage.

distributed over a logarithmic scale. At each resistance value, standby supply voltage is swept from 0 to nominal V_{DD}, i.e., $1.2V$, by increments of $\Delta = 10mV$, and the cell is tested for all drowsy faults. That is: i) we write '1' ('0') to the cell, (ii) switch it to the drowsy mode and pause, and iii) awaken the cell and read. The pass/fail test results of the defective cell at each resistance and standby voltage are presented in the shmoo plot of Figure 3.1(b), as an example. Similar shmoo plots can be obtained for other defects.

The results show that at a given standby voltage, the defective cells can still operate properly, if their defect resistance is below a critical level. For example, at $V_{DDL} = 0.8V$, cells with an open defect, where $R_{open} \leq 1M\Omega$, will still operate properly, while all the cells with $R_{open} > 1M\Omega$ will fail. The critical resistance of each defect is an increasing function of the standby voltage (V_{DDL}). This means that at low standby voltages, even cells with a small defect will fail. The plot also shows that for $R_{open} > 100M\Omega$, the defect becomes a hard defect causing a failure, i.e., data retention fault (DRF), even in the active mode. As can be expected, at $V_{DDL} < DRV = 220mV$ all cells, including healthy cells, fail.

According to the above analysis, if a chip with N_{cells} SRAM cells is tested to be free of strong faults, i.e., no faulty behavior in the active mode, it does not contain defective cells with a resistance value larger than a certain threshold, e.g., $100M\Omega$. However, it is still quite possible that the chip contains some cells with weak defects, i.e., defects with a resistance smaller than a certain value, e.g., $100M\Omega$, which have escaped tests. However, if the chip is switched to drowsy mode, these weak cells may exhibit drowsy faults depending on the level of standby voltage and the resistance value of their defect. As can be seen in Figure 3.1(b), the cumulative number of failing cells increases with the reduction of the standby voltage. Hence, larger number of weak cells will fail as the standby voltage is lowered for larger leakage reductions. This establishes a tradeoff between leakage reduction and yield of drowsy SRAMs.

3.1.2 Wake-up Time

To ensure a fault-free operation, the supply voltage and the voltage of storage nodes of SRAM cells should be restored to their nominal values, before a read/write operation. When a cell is switched from drowsy to active mode, it takes a certain time, called its wake-up time, for all the nodes to restore their nominal voltages. To avoid faults, drowsy memory cells should be woken-up a certain number of clock cycles, called wake-up latency, before the next access.

The wake-up time of defective cells can be much longer than that of healthy cells. For instance, the simulation results for the voltage of storage nodes, i.e., T and F, of a healthy cell and a defective cell with a resistive open defect in the pull-up path of the cell (see Figure 3.1(a)), where $R_{OC1} = 1M\Omega$, are shown in Figure 3.2. For the healthy cell, the wake-up time, i.e., the time required for storage nodes to restore their nominal values, is less than $2ns$. Whereas that of the defective cell is considerably larger ($\sim 10ns$). As mentioned earlier, a wake-up latency of 1-2 clock cycle(s) is

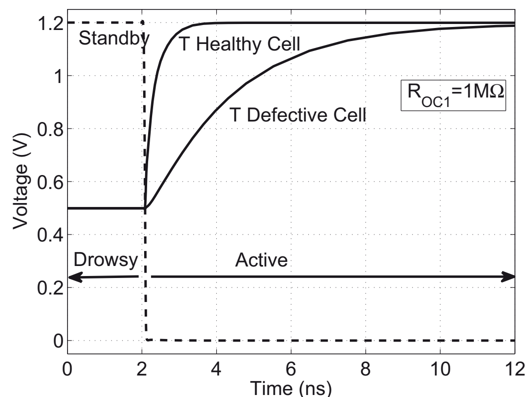


Figure 3.2. Simulation results showing the difference in wake-up time of a healthy cell and a defective cell.

usually considered for drowsy SRAMs, which can usually be hidden in the memory access cycle without incurring any performance penalty [22]. Thus, if such a defective cell with a wake-up time longer than 1-2 clock cycle(s) is accessed immediately after wake-up, there will be a chance of failure due to the unsettled internal voltages. To avoid failures, a larger wake-up latency may be considered. However, it can also result in a remarkable performance penalty. [22]. Thus, the most efficient way to address these failures, which can frequently happen in drowsy SRAMs, is to detect and replace them with spare resources.

3.2 Simulation Methodology

In this work, we performed fault injection and simulation for SRAMs to investigate their behavior when switched to drowsy mode. To do this, we first designed a 6T SRAM cell using standard threshold voltage (SVT) transistors from STM 90-nm technology. Minimum feature size transistors are used for pull-up PMOS and access NMOS transistors (see Figure 3.1(a)). For pull-down NMOS transistors, minimum length is used. However, their width is set so that a cell ratio and write ratio of 2 and

1 are obtained, respectively, i.e., $Wn/Wa = 2$, and $Wp/Wa = 1$, where Wn, Wa, and Wp, are the width of pull-down, access, and pull-up transistors.

We created a 2Kb SRAM block (64 rows by 32 columns) by replicating the designed sram cell. The cells of each column share the precharge, column multiplexer, write driver, and sense amplifier circuits. The simulation setup is shown in Figure 3.3. To account for the parasitics of the power grid network, we modeled it as lumped resistor, inductor, and capacitance elements. Typical values of $R = 30\Omega$, $L = 3nH$, and $C = 10pF$ are assumed for the power grid parasitics [49]. The wire capacitance of the virtual V_{DD} node ($VVDD$), wordlines, and bitlines is estimated as $500fF$, $10fF$, and $30fF$ respectively by performing a netlist extraction of the 2Kb SRAM layout.

During the normal operation, the *Standby* signal is low and thus nominal V_{DD} is applied to the memory block through the PMOS transistor P_{sleep}. By asserting the *Standby* signal, the NMOS transistor N_{sleep} is turned on and the reduced standby voltage V_{DDL} is applied to the block. The P_{sleep} transistor should be sized large enough to avoid write-time penalties and also to achieve a fast wake-up time. During a write operation, a row of cells are selected by the corresponding WL signal and then the data is written to the cells. In the worst-case scenario, where the state of all the cells need to be flipped, P_{sleep} should provide current for the pull-up PMOS transistors of all 32 cells in a row. Thus we generously size the P_{sleep} transistor as 10 times the lump size of all the pull-up transistors, i.e., $10 \times 32 \times 0.1\mu m \approx 30\mu m$. The N_{sleep} transistor can be sized much smaller as it only requires to supply the data-retention current of cells during the standby mode. We set the size of N_{sleep} transistor to $3\mu m$.

We injected defects to one of the SRAM cells in the block and performed HSPICE simulations to identify the faulty behaviors which are sensitized by the drowsy operating mode. In the following, we first describe the defect space and fault modeling

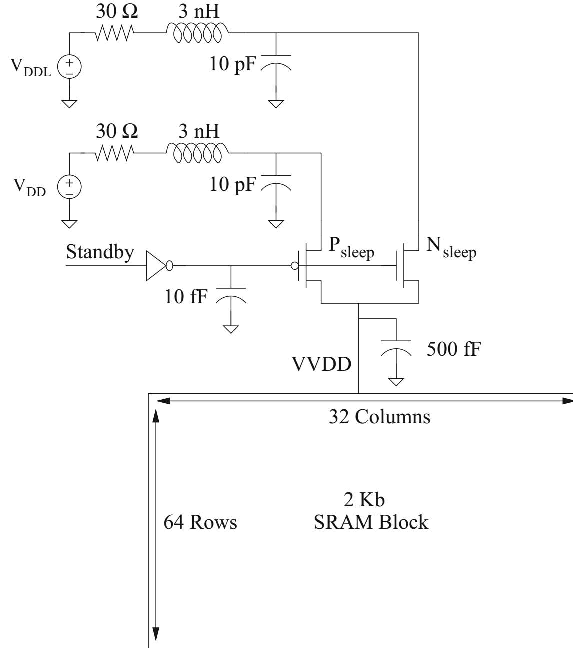


Figure 3.3. Simulation setup.

methodology that we have used in this work. Then, newly observed fault behaviors are explained in detail. It should be noted that our simulations show that different sizing of SRAM cell transistors does not alter the fault injection and simulation results qualitatively, and the reported faults are observed at all sizing. However, the resistance ranges of defects and the level of standby voltage at which a fault is sensitized vary quantitatively.

3.3 Fault Modeling and Notation

In 6-transistor (6T) SRAM cells, most of the short defects, even with a large parasitic resistance, alter the read/write operation of the cell, and thus are usually detected by the traditional March tests. However, detection of open defects in 6T SRAM cells is known to be a challenging and time-consuming task [50, 35]. Resistive-open defects are known to be a major cause of weak faults in SRAMs which tend to

escape traditional March tests [51, 52, 35, 50]. Two categories of open defects in SRAM cells are undetectable using only traditional March tests. The first category includes opens that cause data retention faults (DRFs). The second category of open defects result in SRAM stability degradation causing stability faults [35, 50]. These degraded SRAM cells, called weak cells [35], can usually function properly in normal operating conditions, and thus, no faulty behavior emerges during a regular March test. However, under adverse conditions, i.e., the conditions contributing to some stability degradation, such as a reduced supply voltage during standby mode, these cells may malfunction. Therefore, in this work, we will focus only on resistive-open defects, which exhibit no faulty behavior in normal mode of operation, while they cause faulty behavior with the introduction of a drowsy mode to the memory. The significance of resistive-open defects has considerably increased in recent technologies, due to the large number of interconnect layers and a growing number of connections between them. We adopt the fault modeling methodology presented in [46] in order to experimentally analyze the faulty behaviors that can be caused by open defects in drowsy SRAM cells.

3.3.1 Open Defects in SRAM Cells

All possible open defects in an SRAM cell, denoted as OC, are shown in Figure 3.4. In this circuit diagram, each branch is labeled by a potential resistive open defect by the notation OCx and OCxc, where x denotes the node number. Due to the symmetric structure of the SRAM cell, opens at locations OCx and OCxc will show a complementary fault behavior [46]. Therefore, we consider only opens at OCx locations and test for both ‘1’ and ‘0’ initial conditions of stored data.

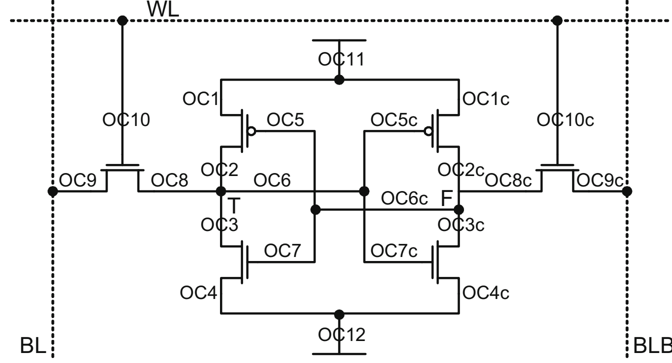


Figure 3.4. All possible open defects in a 6T SRAM cell.

3.3.2 Functional Fault Models

Functional faults are defined as the deviation of the observed memory behavior from the expected one under an operation sequence [53]. Functional fault models (FFMs) e.g., stuck-at faults, data-retention faults, are defined to describe fault behavior of SRAMs [53]. Fault primitives (FPs) mathematically define FFMs by specifying: (1) a sensitizing operation sequence (S), and (2) the observed faulty behavior. FPs can be classified according to the number of different cells accessed by a S, i.e., $\#C$, and according to the number of different operations performed in an S, i.e., $\#O$, [53].

Depending on $\#C$, FPs are divided into the following classes:

- Single-cell FP: If $\#C = 1$, then the FP sensitized by the corresponding S is called a *single-cell FP*.

- Coupling FP: If $\#C > 1$, then the FP sensitized by the corresponding S is called a *coupling FP*.

Depending on $\#O$, FPs are divided into the following classes:

- Static FP: If $\#O = 1$, then the FP sensitized by the corresponding S is called a *static FP*.

- Dynamic FP: If $\#O > 1$, then the FP sensitized by the corresponding S is called a *dynamic FP*.

In this work, we investigate only single-cell FPs, i.e., $\#C = 1$. Dynamic faults are faults sensitized with more than one operation (i.e., $\#O > 1$), and thus there is an infinite number of them. However, it has been shown that the probability that some defects can only be sensitized with large values of $\#O$ is very low, and two-operation dynamic faults are the most popular faults in state-of-the-art memories [53]. Therefore, we restrict ourselves to the case where at the most two operations in sequence are required to catch a defect, i.e., $\#O = 2$.

3.3.3 Fault Notation

We adopt the notation presented in [54] and [53] to describe FPs, and a similar naming convention to describe FFMs. Each FP represents a certain fault behavior and is denoted as $\langle S/F/R \rangle$. Here, S is the sensitizing operation sequence, and is composed of one or more operations which are sequentially performed on an SRAM cell to sensitize a fault. F denotes the data value of the faulty SRAM cell after applying the S to it. There are some faults, however, in which the cell's data is not altered, whereas the value read out from the cell is incorrect. Hence, R is used to denote the read-out value from the cell, in case the last operation in S is a read operation. Thereby, the observed faulty behavior is denoted by F and R collectively.

In the FP notation, S is a sequence of operations which belong to $\{0, 1, w0, w1, r0, r1, dr0, dr1, \forall\}$, where 0 (1) denotes a zero (one) logic value, $w0/w1$ and $r0/r1$ denote write and read operations. $dr0$ ($dr1$) describes a drowsy operation, i.e., switching a cell to the drowsy mode and back to the active mode, with a data '0' ('1'). $\forall$ denotes any operation. For instance, $dr1r1$ denotes that a cell with data value '1' is switched to drowsy mode and then is woken-up, and a read 1 operation is

performed on it immediately after wake-up. Since the duration of time that the cell spends in the drowsy mode is not crucial to sensitize the fault, it is not indicated in the notation. However, a minimum time is still required to allow the cell to reach its early drowsy state before wake-up. If a certain duration of time is essential for sensitizing the fault it is denoted with a subscript P in the notation of the operation. For example, $dr1_P$ denotes that a cell with data value '1' is switched to drowsy mode and is kept in that mode for period P.

Similarly, F denotes the data value of the faulty cell after applying S to it. $F \in \{0, 1, X\}$, where 0 (1) denotes a zero (one) logic value, and X denotes an undefined logic value.

Finally R denotes the output value of the cell in case the last operation in S is a read operation. $R \in \{0, 1, X, -\}$, where '0' ('1') denotes a zero (one) read-out value, X denotes an unknown logic value, and '$-$' is used when the output data is either not applicable or don't-care. E.g., if $S = dr1w1$, then no data is read out from the memory, and thus R is not applicable.

3.4 SRAM Drowsy Faults Due to Resistive-Open Defects

We performed fault injection and simulation for each of the open defects shown in Figure 3.4. Each open defect is modeled as a resistance with a value logarithmically distributed over the 0 to $1T\Omega$ range, incrementing as 100, $1K$, $10K$, etc. As will be shown later, the applied standby voltage determines if a defect, with a certain resistance value, will result in a faulty behavior. Thus, we set the standby voltage as low as $V_{DDL} = 0.3V$, which is slightly above the DRV of healthy cells, so that all the potential faults are sensitized. At each resistance value, all possible operations,

Table 3.1. Single-Cell Static and Dynamic Faults In Drowsy SRAM Due To PODs.

Defect Location	Resistance	Fault Behavior	Comp. Behavior	Type	FFM
OC1, OC2	I	wF	wF	–	–
	II	$\langle dr1r1/0/-\rangle$	$\langle dr0r0/1/-\rangle$	Dynamic	Drowsy Read Destructive Fault (DRDF)
		$\langle dr1r1/0/1\rangle$	$\langle dr0r0/1/0\rangle$	Dynamic	Drowsy Deceptive Read Destructive Fault (DDRDF)
		$\langle dr1r1/1/0\rangle$	$\langle dr0r0/0/1\rangle$	Dynamic	Drowsy Incorrect Read Fault (DIRF)
	III	$\langle dr1/0/-\rangle$	$\langle dr0/1/-\rangle$	Static	Drowsy Transition Fault (DTF)
		$\langle dr1_P/0/-\rangle$	$\langle dr0_P/1/-\rangle$	Static	Drowsy Data Retention Fault (DDRF)
		$\langle 1_P/0/-\rangle$	$\langle 0_P/1/-\rangle$	Static	Data Retention Fault (DRF)
OC5	I	wF	wF	–	–
	II	$\langle dr1r1/0/-\rangle$	$\langle dr0r0/1/-\rangle$	Dynamic	Drowsy Read Destructive Fault (DRDF)
		$\langle dr1r1/1/0\rangle$	$\langle dr0r0/0/1\rangle$	Dynamic	Drowsy Incorrect Read Fault (DIRF)
	III	$\langle dr1/0/-\rangle$	$\langle dr0/1/-\rangle$	Static	Drowsy Transition Fault (DTF)
		$\langle dr1_P/0/-\rangle$	$\langle dr0_P/1/-\rangle$	Static	Drowsy Data Retention Fault (DDRF)
		$\langle 1_P/0/-\rangle$	$\langle 0_P/1/-\rangle$	Static	Data Retention Fault (DRF)
OC11	I	wF	wF	–	–
	II	$\langle dr1r1/0/-\rangle$	$\langle dr0r0/1/-\rangle$	Dynamic	Drowsy Read Destructive Fault (DRDF)
		$\langle dr1r1/1/0\rangle$	$\langle dr0r0/0/1\rangle$	Dynamic	Drowsy Incorrect Read Fault (DIRF)
		$\langle dr1r1/0/1\rangle$	$\langle dr0r0/1/0\rangle$	Dynamic	Drowsy Deceptive Read Destructive Fault (DDRDF)
		$\langle dr1r1/X/-\rangle$	$\langle dr0r0/X/-\rangle$	Dynamic	Drowsy Undefined State Fault (DUSF)
	III	$\langle dr1_P/X/-\rangle$	$\langle dr0_P/X/-\rangle$	Static	Drowsy Data Retention Fault (DDRF)
		$\langle dr1_P/0/-\rangle$	$\langle dr0_P/1/-\rangle$		
		$\langle dr1/0/-\rangle$	$\langle dr0/1/-\rangle$	Static	Drowsy Transition Fault (DTF)
		$\langle 1_P/X/-\rangle$	$\langle 0_P/X/-\rangle$	Static	Data Retention Fault (DRF)
		$\langle 1_P/0/-\rangle$	$\langle 0_P/1/-\rangle$		

where $\#C = 1$ and $\#O \leq 2$, are examined in the presence of an open defect. Our simulations show that a drowsy operation sensitizes only faults due to open defects at locations: OC1, OC2, OC5, and OC11. At large parasitic resistances, these defects result in a failure in both active and drowsy modes. However, at lower resistance values, they start to cause failures only when the memory is switched to drowsy mode. The other open defects, depending on their resistance value, either cause a failure in active mode, or do not cause a failure at all, i.e., they are not sensitive to supply voltage reduction. The aforementioned open defects are all related to the pull-up PMOS devices, thus we refer to them as PMOS open defects (PODs) [50].

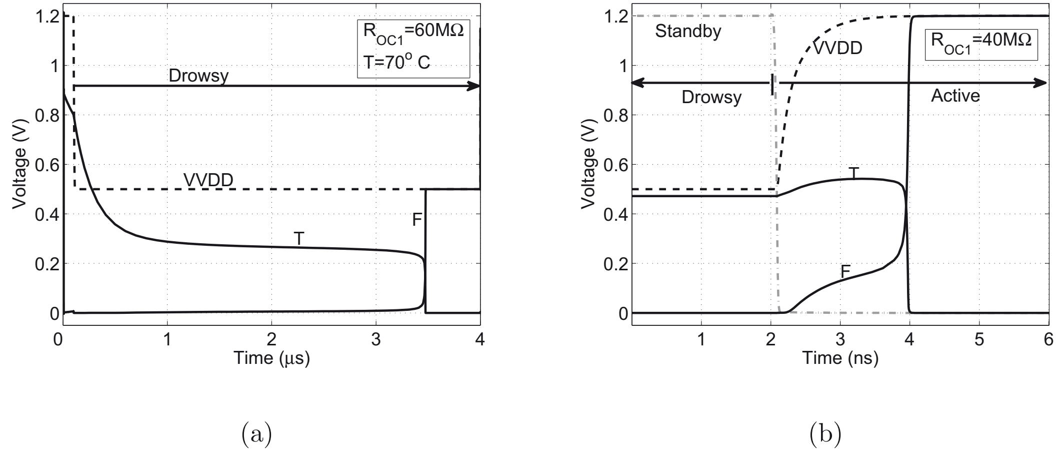
Simulation results of single-cell static and dynamic FFMs, where at least one operation in S is a drowsy operation, are listed in Table 3.1. We list only the defects which have resulted in a new faulty behavior when the cell is switched to drowsy mode. Hence, this table summarizes all the observed new drowsy faults for each

POD. The first column in this table gives the location of the defect. Simulation results show that (OC1, OC2) exhibit identical fault behavior when at the same resistance regions. Thus, they are listed on the same row in Table 3.1. The second column lists the resistance regions in increasing order of resistance value (from 0 to $1T\Omega$). Note that the resistance regions are not identical for all defects. For example, the resistance region II of defect (OC1, OC2) and OC11 is from $100K\Omega - 1M\Omega$, while that of defect OC5 is from $200M\Omega - 1G\Omega$. The third and fourth columns give the fault behavior and the complementary fault behavior of each defect, respectively. The complementary fault behavior of a defect OCx is the fault behavior caused by an OCxc defect (see Figure 3.4). Each faulty behavior is reported in terms of a fault primitive (FP), if a strong fault is sensitized. FP notation presented in Subsection 3.3.3 is used to describe a strong fault. If a fault is only partially sensitized, then it is denoted as a weak fault (wF). The fifth column classifies the sensitized fault as static or dynamic. Finally, the FPs are translated into FFMs and are listed in the last column of Table 3.1.

The FFMs are divided into static and dynamic FFMs and are described in the following.

3.4.1 Static Drowsy Faults (SDF)

New static fault behaviors appear in the memory due to the introduction of the drowsy mode. For example, when a cell with an OC1 defect (see Figure 3.4) and a parasitic resistance $R_{OC1} = 60M\Omega$, is switched to the drowsy mode, the state of the cell flips after a certain time. The circuit simulation results for this particular fault are shown in Figure 3.5(a). The initial logic value of the cell is '1', however due to the OC1 defect, T is around $0.8V$. At $t = 100ns$, the cell is switched to the drowsy mode by lowering the virtual V_{DD} node ($VVDD$) to $V_{DDL} = 0.5V$. Node T starts to follow



(a) (b)

Figure 3.5. HSPICE simulation results of a defective cell exhibiting (a) a drowsy data-retention fault (DDRF) and (b) a drowsy transition fault (DTF).

$VVDD$, however it continues to decrease below $0.5V$ and eventually at $t \approx 3.5\mu s$ the state of the cell flips (T falls to zero and F rises to $0.5V$).

This fault is a data retention fault, which is caused by the leakage currents of NMOS transistor MnL (see Figure 3.1(a)). In a healthy cell, the on current of pull-up PMOS transistor MpL is much larger than the leakage of MnL, and thus node T is kept charged all the time. However, if the on current of MpL is diminished due to an OC1 defect, such that it can no longer compensate for the leakage currents of MnL, node T gradually discharges to ground.

To sensitize this fault, the defective cell, with the initial value '1', should be switched to the drowsy mode and kept in that mode for a certain period (P). Thus, the sensitizing operation sequence for this fault can be expressed as $dr1_P$. The subscript P denotes the drowsy time required for this fault to emerge, which is in the range of 100ns-1ms. After the cell spends this period (P) in drowsy mode, its value (F) flips to '0'. Since there is no read operation in S, R is denoted as not applicable in the FP notation, i.e., '-'. Thus, this fault can be represented as $\langle dr1_P/0/-\rangle$

($\langle dr0_P/1/-\rangle$). This FFM, called drowsy data retention fault (DDRF), can be caused by any of the PODs as shown in Table 3.1.

Another new fault behavior is the drowsy transition fault (DTF), where the state of the cell flips when it is woken-up. The simulation results of a cell with an OC1 defect with a parasitic resistance ($40M\Omega$), which exhibits this fault, are shown in Figure 3.5(b). Note that this defective cell exhibits no faulty behavior during active mode. The initial logic value of the cell is '1' ($T = 1.2V$). When it is switched to drowsy mode ($V_{DDL} = 0.5V$), node T falls to a voltage slightly below $0.5V$, and stays at that level during the drowsy period. However, when the cell is woken-up at $t = 2ns$ (see Figure 3.5(b)) by rising the supply voltage to $1.2V$, node T does not rise as expected. Instead, after a short period (at $t \approx 4ns$) the state of the cell flips (T falls to zero and F rises to $1.2V$). This is due to the imbalance in the cell caused by OC1 defect on its left side. Actually, the cell functions as a sense amplifier during wake-up. Although the differential voltage on the two sides of the cell is in favor of node T, the imbalance due to the OC1 defect causes the node F to eventually prevail.

This fault is sensitized by switching the defective cell, with initial value '1', to the drowsy mode and waking it up, hence, $S = dr1$. The duration of the pause in the drowsy mode is not important for sensitizing this fault, thus it is not indicated in the notation. F is denoted as '0' in the FP, which means that the cell flips after the wake-up. There is no read operation in S again, hence R is denoted as '-'. Thus, this fault is represented as $\langle dr1/0/-\rangle$ ($\langle dr0/1/-\rangle$) in Table 3.1, and can be caused by any of the PODs.

To summarize, the following single-cell static FFMs are derived, based on the fault simulation results of PODs:

3.4.1.1 Drowsy Data Retention Fault (DDRF)

A cell is said to have a drowsy data retention fault (DDRF) if it fails to retain its data after spending some period of time P in the drowsy mode. DDRF consists of four FPs: $\langle dr1_P/0/-\rangle$, $\langle dr0_P/1/-\rangle$, $\langle dr1_P/X/-\rangle$, and $\langle dr0_P/X/-\rangle$; and it can be caused by any of the PODs.

3.4.1.2 Drowsy Transition Fault (DTF)

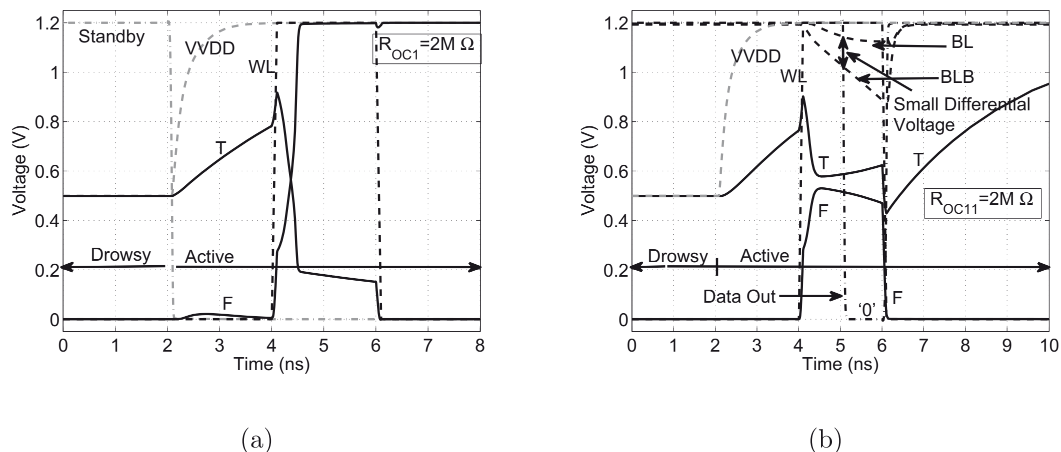
A cell is said to have a drowsy transition fault (DTF) if its data transitions from x to $\bar{x}$, when the cell is woken up. DTF consists of two FPs: $\langle dr1/0/-\rangle$ and $\langle dr0/1/-\rangle$; and it can be caused by any of the PODs.

3.4.2 Dynamic Drowsy Faults (DDF)

As discussed in Section 3.1, due to the frequent switching of SRAM cells between active and drowsy modes in drowsy caches, dynamic faults can potentially occur due to a read/write operation performed immediately after (before) a transition from (to) the drowsy mode. The above mentioned faults involve more than one operation in sequence, thus they are dynamic faults. As the sensitizing operation includes a drowsy operation, we call them dynamic drowsy faults (DDF). There are four types of 2-operation sequences involving a drowsy operation:

1. $w1dr1$ $(w0dr0)$: A drowsy operation performed immediately after a write operation.

2. $r1dr1$ $(r0dr0)$: A drowsy operation performed immediately after a read operation.



(a) (b)

Figure 3.6. (a) Simulation results of a defective cell exhibiting (a) a drowsy read-destructive fault (DRDF) and (b) a drowsy incorrect read fault (DIRF).

3. $dr1w0$ ($dr0w1$) : A write transition operation performed immediately after wake-up.

4. $dr1r1$ ($dr0r0$) : A read operation performed immediately after wake-up.

We performed fault simulations of all POD defects, with a parasitic resistance logarithmically changing from 100Ω-$1T\Omega$, i.e., 100, $1K$, $10K$, etc., with the above operation sequences. The simulation results are compiled in Table 3.1. The results show that only a read operation performed immediately after wake-up is resulting in new fault behaviors. No faulty behaviors were observed when a cell is switched to drowsy mode immediately after a read/write operation. Neither, a write operation after wake-up causes a fault. This is due to the fact that only a read operation performed immediately after wake-up further disturbs a defective cell while it is in a transitional mode, causing it to fail.

New fault behaviors appear when performing a read operation immediately after wake-up. The simulation results for an OC1 defect (see Figure 3.4) with a parasitic resistance ($R_{OC1} = 2M\Omega$), which exhibits a new fault behavior, are shown in Figure 3.6(a). At nominal supply voltage, i.e., $1.2V$, this cell exhibits no faulty behavior.

Initially, logic value '1' is written to the cell. Then, it is switched to drowsy mode by lowering the voltage of virtual V_{DD} node to $V_{DDL} = 0.5V$. The cell retains its data when in the drowsy mode, and there is no drowsy transition fault (DTF) when it is woken-up. However, when a read operation is performed on the cell $2ns$ after wake-up, i.e., at $t = 4ns$, the cell flips.

As can be seen from the waveforms of Figure 3.6(a), after wake-up, voltage of node T does not rise to $1.2V$ as fast as $VVDD$ due to the OC1 defect. Thus, when the cell is accessed at $t = 4ns$, i.e., one clock cycle after wake-up, it is still in the midst of transition from drowsy to active mode. This makes the cell very vulnerable to the extra disturbance applied by the read operation. Therefore, the cell flips and a wrong logic value is read out.

This fault is represented as $\langle dr1r1/0/-\rangle$ ($\langle dr0r0/1/-\rangle$) in Table 3.1. Here, $S = dr1r1$, which means that a read 1 operation should be performed on the defective cell, with initial value '1', immediately after wake-up. This causes the cell to flip to '0' ($F =$'0'). The read-out value is not important, hence $R =$'-'. This fault model, called a drowsy read destructive fault (DRDF), can be caused by any of the PODs as can be seen in Table 3.1.

Another newly observed fault behavior is caused by a read operation after wake-up. This new behavior, called drowsy incorrect read fault (DIRF), happens when a read operation returns an incorrect value without flipping the cell's state. The simulation results of a cell with an OC11 defect with a $2M\Omega$ parasitic resistance, which exhibit this fault, are shown in Figure 3.6(b). The cell successfully retains its initial logic value, i.e., '1', during drowsy mode, and after wake-up at $t = 2ns$,. Even accessing the cell at $t = 4ns$ does not cause it to flip, and the cell starts to restore its correct logic value after the access is over at $t = 6ns$. However, an incorrect value is read out, i.e., $Data_Out = 0$.

To ensure a correct read out by sense amplifiers in all circumstances, a differential voltage larger than a certain value, e.g., 10% of nominal V_{DD}, must form between BL and BLB when the sense operation is triggered. In this example, the OC11 defect prevents the voltage of node T from rising to $1.2V$ as fast as VV_{DD}. Thus, when the cell is accessed at $t = 4ns$, the voltage of node T is below $1.2V$, causing a discharge of BL. In addition, during the access cycle, the voltage of node F rises, slowing the discharge of BLB. Consequently, not enough differential voltage is developed at the end of the read cycle (see Figure 3.6(b)) between BL and BLB. Eventually, this small differential voltage results in an incorrect read out due to the imbalance in the sense amplifier caused by process parameters variations.

This fault is represented as $\langle dr1r1/1/0\rangle$ ($\langle dr0r0/0/1\rangle$) in Table 3.1. To sensitize this fault a read 1 operation immediately after wake-up should be performed on the defective cell, with initial value '1', i.e., $S = dr1r1$. The read operation returns an incorrect value ($R =$'0'), although the cell's data does not flip ($F =$'1').

To summarize, the following new single-cell dynamic FFMs are derived based on the simulation results of PODs:

3.4.2.1 Drowsy Read Destructive Fault (DRDF)

A cell is said to have a drowsy read destructive fault (DRDF) if a read operation performed to a cell with value x immediately after wake-up changes the cell's data to $\bar{x}$ and returns the logic value $\bar{x}$. DRDF consists of two FPs: $\langle dr1r1/0/-\rangle$ and $\langle dr0r0/1/-\rangle$.

3.4.2.2 Drowsy Incorrect Read Fault (DIRF)

A cell is said to have a drowsy incorrect read fault (DIRF) if a read operation performed to a cell with value x immediately after wake-up returns the logic value

$\bar{x}$ while the cell's data remains at x. DIRF consists of two FPs: $\langle dr1r1/1/0\rangle$ and $\langle dr0r0/0/1\rangle$.

3.4.2.3 Drowsy Deceptive Read Destructive Fault (DDRDF)

A cell is said to have a drowsy deceptive read destructive fault (DRDF) if a read operation performed to a cell with value x immediately after wake-up changes the cell's data to $\bar{x}$ but returns the logic value x. DDRDF consists of two FPs: $\langle dr1r1/0/1\rangle$ and $\langle dr0r0/1/0\rangle$.

3.4.2.4 Drowsy Undefined State Fault (DUSF)

A cell is said to have a drowsy undefined state fault (DUSF) if accessing it immediately after wake-up changes its state to an unknown state (X). DUSF consists of two FPs: $\langle dr1r1/X/-\rangle$ and $\langle dr0r0/X/-\rangle$. It was observed that this fault can be caused only by OC11.

3.5 Testing for Drowsy Faults

In Section 3.4, the existence of dynamic drowsy faults has been validated using HSPICE simulations, and FPs were derived for the new drowsy faults. In this section, we use the derived FPs to design a March test for detection of the newly observed faults. Authors in [54] propose a March algorithm to detect static drowsy faults in a word-oriented memory. However, they completely ignore dynamic drowsy faults. Here, we develop a March test which is able to detect all drowsy faults as well as the traditional faults in SRAMs.

Table 3.2. March RAD Test.

$\{\Downarrow(w0);$	$\Uparrow(r0, w1, r1, w0);$	$dr_P;$	$\Downarrow(r0, r0);$
M_1	M_2	M_3	M_4
$\Uparrow(w1);$	$\Downarrow(r1, w0, r0, w1);$	$dr_P;$	$\Uparrow(r1, r1);\}$
M_6	M_7	M_8	M_9

3.5.1 March RAD

To permit the detection of drowsy faults we propose to insert drowsy elements to traditional March tests. In this work, we extend the March SR test [46] to cover the drowsy faults introduced in Section 3.4. March SR test covers all simple realistic faults discussed in [46], hence, the proposed new test will automatically detect them as well. As it was shown, dynamic drowsy faults happen due to a read-after-drowsy operation, i.e., read operation immediately after wake-up. Detection of these faults requires writing a certain data to the cell, switching it to drowsy mode, and thereafter reading the cell immediately after wake-up. These steps have to be done for both logic states of the cell, i.e., '0', and '1'. The new March algorithm, referred to as March RAD ("read-after-drowsy"), achieves this through two newly inserted drowsy operations as shown in Table 3.2.

We use the traditional March notation in order to describe March RAD test. A complete March test is delimited by a pair of brackets '{...}', while a March element is delimited by a pair of parentheses '(...)'. The March elements are separated by semicolons, and the operations within a March element are separated by commas. All operations of a March element are performed at a certain address, before proceeding to the next address. This can be done in either one of two address orders: an increasing ($\Uparrow$) or a decreasing ($\Downarrow$) address order. When the address order is not relevant, the symbol $\Updownarrow$ will be used. The drowsy operation (dr_p) in the proposed test, means that the whole memory is put in drowsy mode for a period p, which is the longest

time required for activation of data-retention faults. To detect a read-after-drowsy fault within a memory line, it is imperative that a read operation is performed on it immediately after it is woken-up. Therefore, in the proposed test, March elements M_4 and M_9 are performed on memory lines by first awakening only the corresponding line and then performing a read operation on it. Thereby, the memory is woken-up one line at a time by M_4 and M_9, so that the whole memory will be in active mode when these elements complete. This requirement is further discussed for different drowsy cache architectures in the next section.

Table 3.3 shows by which March elements (i.e., M_1 through M_9) of March RAD, each FP belonging to each single-cell drowsy FFM, is sensitized and detected. The third column shows the operation that sensitizes the fault and the fourth column shows the operation that will detect it. The fault coverage of March-RAD test can be summarized as follows:

- All SDs which have FPs in active mode will be detected by March RAD, because it contains all the March elements of March SR test [46].
- All DDRFs, DTFs, and DIRFs are detected since a '0' and '1' is read from each single cell (by M_4 and M_9) after a drowsy operation (M_3 and M_8).
- All DRDFs and DDRDFs are detected because a '0' and '1' is read twice consecutively from each single cell (by M_4 and M_9) after a drowsy operation (M_3 and M_8).

All FFMs with a deterministic data output at the sense amplifier can be detected by the proposed test. However, the drowsy FFMs with a random data output may probabilistically be detected as each cell is read with different data values by March elements M_4 and M_9. It should be noted that, for detection of faults such as DUSF which can result in an undefined output, i.e., an output voltage between high and low

Table 3.3. March RAD Fault Coverage.

FFM	FP	Sensitizing	Detecting
DDRF	$\langle dr0_P/1/-\rangle$	M_3	M_4
	$\langle dr1_P/0/-\rangle$	M_8	M_9
DTF	$\langle dr0/1/-\rangle$	M_3	M_4
	$\langle dr1/0/-\rangle$	M_8	M_9
DIRF	$\langle dr0r0/0/1\rangle$	M_3	M_4
	$\langle dr1r1/1/0\rangle$	M_8	M_9
DRDF	$\langle dr0r0/1/-\rangle$	M_3	M_4
	$\langle dr1r1/0/-\rangle$	M_8	M_9
DDRDF	$\langle dr0r0/1/0\rangle$	M_3	M_4
	$\langle dr1r1/0/1\rangle$	M_8	M_9

threshold, a voltage-window-detection circuit similar to one proposed in [54], needs to be used. The March RAD test contains 9 March elements and has a test length of $14n + 2 \times Drowsy$, where n denotes the number of memory locations and $Drowsy$ denotes a drowsy operation on the whole memory. Thus, the proposed test will have a longer test time compared to the traditional March tests, due to the extra test time required for covering drowsy faults.

3.5.2 Test Implications of Drowsy Cache Architectures

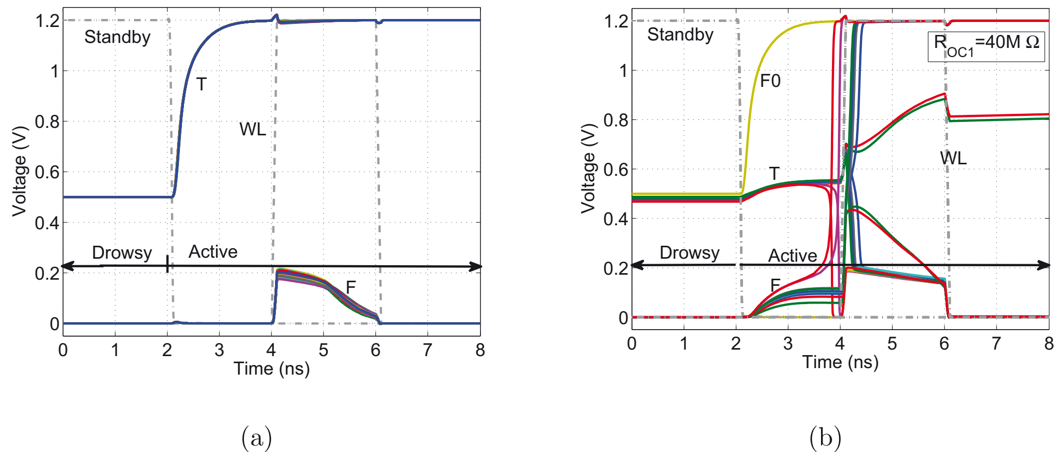
As discussed in Chapter 2, the drowsy design technique has been applied to caches at two different granularities: i) word-oriented and ii) subarray-oriented. In word-oriented drowsy caches [19, 21], every word has its own wake-up signal and thus can be independently awakened. While, in subarray-oriented caches [22], the wake-up signal is shared by all the memory words within a certain sub-array. The proposed March-RAD test requires that each individual line in the memory can be awakened independent of the other lines. For a word-oriented drowsy memory, the proposed March-RAD test functions by putting the whole memory in drowsy mode at once, i.e.,

by dr_p element, and then awakening every word one-by-one just before it is accessed. However, in a subarray-oriented memory, all the lines within a subarray are awakened at once when an access is issued to a line residing within that particular subarray. Hence, to sensitize read-after-drowsy faults at all memory lines, it is necessary that a subarray is switched back to drowsy mode after every access. This can result in an unacceptably large test time due to frequent switching between active and drowsy modes. Thus, a word-oriented drowsy control of the memory is imperative to reduce test time for dynamic drowsy faults using the proposed March-RAD test. This may require additional circuitry in subarray-oriented memories to enable a word-by-word control of switching between active and drowsy modes during test time.

3.5.3 Sensitivity to Process Parameters Variations

The behavior of SRAM cells against a test procedure can significantly vary by the process parameters variations. It is very important that a test strategy to detect defects is not compromised by the process parameters variations. Therefore, a new test procedure should not only be characterized for its effectiveness against the defects in the presence of process parameters variations, but it should also be characterized to see what kind of additional failures it casues by probably making the good devices to fail. In fact, if a fault behavior is within the limits of behavior of good cells, a new test that is designed to detect it, will also lead to the failure of some good cells, resulting in an unacceptable yield loss.

We performed Monte Carlo simulations to characterize the behavior of healthy cells, within the 3-sigma process parameters variations, when the proposed March RAD test is performed on them. The results of the simulations for 100 Monte Carlo points are shown in Figure 3.7(a). As can be seen, all the good cells behave correctly



(a)

(b)

Figure 3.7. (a) Monte Carlo simulation results of (a) a healthy cell and (b) a defective cell with $R_{OC1} = 40M\Omega$, when the March RAD test is performed on them.

and no faulty behavior is observed. This indicates that the proposed test does not result in yield losses due to the rejection of good cells.

To characterize the effectiveness of the test against defective cells in the presence of process parameters variations, we performed a limited number of MC simulations on a defective cell as well. The important observation is that cells with the same defect exhibit different fault behaviors due to the variations in their device parameters. For example, the results for 10 Monte Carlo simulations are shown in Figure 3.7(b). As can be seen, cells exhibit different behaviors when they are woken-up and a read operation is performed on them. In this example, one of the cells, denoted by $F0$, exhibits a read destructive fault during the normal operating mode. This fault can be detected by the traditional March tests. Two of the cells exhibit a DTF fault by flipping after wake-up and before read operation. A majority of cells exhibit DRDF fault as they flip when they are read immediately after wake-up. Interestingly, two cells pass the March RAD test and produce correct output data. This indicates that the proposed test is not capable of detecting such defective cells with a marginal faulty behavior. Thus, a more complex test procedures needs to be developed for their detection.

3.6 Summary

In this chapter, we showed that some spot defects (SDs) in SRAM cells can result in new fault behaviors during the drowsy mode, while they only cause weak faults in the active operating mode and thus escape the traditional March tests. Open defects are known to be a major source of test escapes in SRAMs. Hence, we performed fault injection and simulation to investigate the fault behavior of open defects when an SRAM cell is switched to drowsy mode. It was observed that PMOS open defects (PODs) are a major potential source of test-escapes in the active mode which can cause faults when the memory is switched to the drowsy mode. We extracted fault primitives (FPs) for the newly observed drowsy faults. Then, we used the derived FPs to design a March test for detection of the newly observed faults. The proposed March test, called March RAD, is capable of detecting all drowsy faults as well as the traditional faults in SRAMs. Finally, it was observed that the level of standby voltage determines whether a resistive open defect in an SRAM cell causes a fault in drowsy mode. In general, as the supply voltage is reduced to cut down more leakage, larger number of defects are sensitized, resulting in more failing cells within a memory array. This establishes a trade-off between leakage reduction and yield of SRAMs.

Chapter 4

Aggressive Leakage Reduction of SRAMs Using Fault-Tolerance Techniques: The Yield-Power Tradeoff

Turning down the circuits' knobs during the standby modes to reduce the leakage of memories also results in the reduction of the bitcells' robustness. Therefore, the main goal while attempting to reduce the leakage power of SRAMs is to limit the yield losses due to the cells within the memory array that fail to retain their data reliably. In this chapter, we develop analytical models to analyze the involved yield-leakage tradeoffs in SRAMs. Due to the importance of rare failure events in large memories, an accurate model for the tail of the cell failure probability distribution is developed based on concepts from extreme value theory (EVT) [55, 47]. The efficiency of various fault-tolerance techniques to enhance the leakage reductions while preserving a high yield are also investigated. The analysis is performed using source-biasing as the

leakage reduction technique. However, the analysis and the techniques are equally applicable when other leakage reduction techniques are employed.

4.1 Maximum Applicable Source-Bias Voltage to a Memory

Source-biasing is one of the efficient leakage reduction techniques, and thus is adopted in modern commercial microprocessors such as Intel's Xeon® processor [22]. However, source-biasing also results in the reduction of the cells noise margin [33, 19], making them very vulnerable to data-retention failures (DRFs) [40, 56, 34]. Thus, there is a maximum source-bias voltage (V_{SBmax}) that can be applied to a cell without destroying its data content. Due to the process parameters variations, V_{SBmax} of different cells varies within a die. For example, the histogram of V_{SBmax} obtained by 5000 Monte Carlo (MC) simulations in a 45-nm technology is shown in Figure 4.1. The maximum applicable source-bias voltage to such a memory array is determined by the cell which has the smallest V_{SBmax}. Therefore, the lower bound of V_{SBmax} from this histogram can be determined as the maximum V_{SB} that can be applied to this memory [44, 45]. However, for large memory arrays, with millions of replicated bitcells, the rare events resulting in cells with a V_{SBmax} extremely deviated from its typical value should also be taken into account.

A close examination of Figure 4.1 reveals that the V_{SBmax} distribution tends to exhibit a heavy left tail, which is not fully represented due to the limited number of MC simulations. The rare events in the tail of this distribution might not be an issue in small memory arrays, as their probability of occurrence is extremely low. However, the probability of these rare events can be magnified in large memory arrays due to the large number of cells on a single chip [47, 48]. If the applied source-bias voltage

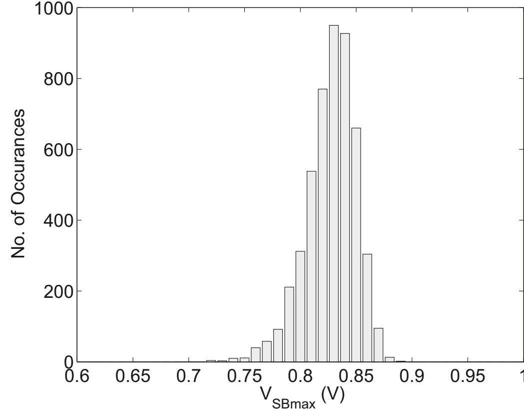


Figure 4.1. Histogram of maximum applicable source-bias voltage to SRAM cells (V_{SBmax}) obtained by 5000 Monte Carlo simulations.

(V_{SB}) to a memory array exceeds V_{SBmax} of these rare event cells, they will fail to retain their data during the sleep mode, resulting in the DRFs. These cell failures, in turn, can cause the whole array to fail, entailing significant yield losses. For example, with a small 64KB memory, even a cell failure probability as low as 1.9×10^{-8} will result in 1% yield loss.

Fault-tolerance techniques can be used to allow for more aggressive leakage reduction in SRAMs by countering DRFs during the sleep mode. Authors in [44, 45] have investigated the fundamental bounds on leakage power reductions in SRAMs, by incorporating error correcting codes (ECCs) to detect and correct the cells which fail in the sleep mode. However, for the statistics of data retention failures, they have used empirical data obtained by measurements from a 4Kb SRAM chip. Hence, the rare failure events have not been taken into account in their analysis due to the limited number of data points. As mentioned before, this can lead to underestimated yield degradations in large memory arrays. Another fault-tolerant leakage reduction technique was proposed in [57], where redundant resources are added to SRAMs to keep the DRFs under control. The obtained leakage reductions, however, are calculated based on the assumption of a normal distribution for the V_{SBmax} of SRAM

cells. Another approach has been presented in [58], where the tail of the V_{SBmax} distribution is assumed to be uniform. This ad-hoc assumption can also cause gross inaccuracies in yield predictions, particularly in the tail regions of the distribution, where the normal distribution significantly deviates from the true tail distribution. In the following, we derive an accurate model for the probability of DRFs as a function of the source-bias voltage using concepts from extreme value theory (EVT) [55, 47]. This enables us to study the tradeoff between leakage reduction and yield of SRAMs when different fault-tolerance techniques are employed.

4.2 Modeling the Tail of the V_{SBmax} Distribution

Accurate estimation of the statistics of extremely rare events using regular MC simulations can be computationally intractable [48]. Recently, two new methods have been proposed in the literature for fast, yet accurate, estimation of rare failure events in SRAMs based on : i) importance sampling [48], and ii) peak over threshold [47] techniques. The results accuracy with the importance sampling technique is very sensitive to the chosen biased sampling distribution [48]. Moreover, special assumptions need to be made about the distribution of process parameters in this technique, limiting its applicability. However, the peak over threshold method does not require an a priori knowledge of the parameters statistics, and provides a complete closed-form model of the tail distribution. Thus, in this work, we use the latter technique to model the tail of the V_{SBmax} distribution.

The peak over threshold method uses concepts from extreme value theory (EVT) [55] to derive a sound distribution model for the exceedances over a high threshold. EVT is a branch of probability theory that studies the statistics of extreme (or rare) events [55]. A seminal result of this theory is that, in most practical cases, a simple analytical generalized Pareto distribution (GPD) can be fitted to the data in the tail

of the distributions, and thereby predictions can be made further out in the tail [55]. The cumulative distribution function (CDF) of GPD with shape parameter ξ and scale parameter *beta* is [55]

$$G_{\xi,\beta}(z) = \begin{cases} 1 - \left(1 - \xi \frac{z}{\beta}\right)^{1/\xi}, & \xi \neq 0; z \in D(\xi, \beta) \\ 1 - e^{-z/\beta}, & \xi = 0; z \geq 0 \end{cases} \tag{4.1}$$

where

$$D(\xi, \beta) = \begin{cases} [0, \infty), & \xi \leq 0 \\ [0, \beta/\xi], & \xi > 0. \end{cases} \tag{4.2}$$

An implementation of the peak over threshold method to model the tail distribution of circuit metrics, such as write time, data-retention voltage etc., is realized in the Statistical Blockade (SB) tool [47, 59]. In this thesis, we use this tool set to model the rare events in the tail of V_{SBmax} distribution. In the SB tool, first a sufficient number of tail data points are generated and then a GPD is fitted to these data. To do this in a reasonable time, the SB tool uses a classifier to filter a very large number of MC points prior to simulation selecting only a subset of them that are likely to appear in the tail of the distribution. The employed classifier is a support vector machine (SVM) [47]. SVMs are supervised-learning classifiers that can take any point from the input space and predict its membership to one of two classes. In the SB tool, a small number of MC simulations are performed and the results are used to train the classifier. Then, a large number of MC points are generated and filtered by the classifier, identifying those that belong to the tail of the distribution. This subset of points are simulated to produce a sufficient number of true tail points. A GPD is finally fitted to the obtained tail data. Details of the simulation setup and the procedure to model the tail of the V_{SBmax} distribution are described in the following sections.

4.2.1 Simulation Setup and Process Variation Model

We use transistor models from 45-nm predictive technology model (PTM) [17] for simulation of a 6T SRAM cell. Since there is no process variation technology file available for the predictive technology models, we use the methodology presented in [60] to model process variations in our MC simulations. In SRAMs, parametric failures are mostly due to the mismatches among transistors in a cell [61]. Indeed mismatches between parameters from distinct cells generally have no incidence on the operation of an SRAM. The primary source of the device mismatches in contemporary technologies is the intrinsic fluctuations in transistors threshold voltage (Vt) due to random dopant fluctuations [40]. Hence, in this work, we have restricted our model only to variations in Vt. The intrinsic variations in Vt have a strong random component of growing significance with advanced processes. Hence, no correlation is considered between variations in Vt of adjacent transistors. Therefore, we have modeled threshold voltage of the transistors in each SRAM cell as six independent Gaussian random variables, generated from two distinct distributions, one for the PMOS and one for the NMOS transistors

$$\begin{aligned} Vtn &\sim N(Vtno, \sigma Vtn) \\ Vtp &\sim N(Vtpo, \sigma Vtp) \end{aligned} \tag{4.3}$$

where $Vtno$ and $Vtpo$ are the nominal threshold voltage of NMOS and PMOS transistors, respectively.

The standard deviations of threshold voltage variation (σVtn and σVtp) depend on the manufacturing process and the size of transistors [61]. Process parameters variation is expected to increasingly deteriorate in nanometer technologies [40]. Moreover, as the minimum size transistors are used in SRAM cells, the mismatches among them are accentuated. For example, σVt as high as $\sim 45mV$ is reported in a 45-nm technology [7] for minimum-geometry devices. Assuming a nominal threshold voltage of

$\sim 300mV$, this accounts for $\sim 15\%$ relative standard deviation, i.e., $\sigma Vt/Vt = 0.15$. In this work, the analysis is performed assuming $5\% - 10\%$ variation in Vt. The temperature is set to $T = 70°C$ in all simulations.

4.2.2 Tail Modeling Procedure

The peaks over threshold method is developed to fit a GPD to the right tail of the distributions. However, the V_{SBmax} distribution has a left tail as shown in Figure 4.1. Hence, we convert the left tail of the distribution to a right tail by replacing $\hat{V}_{SBmax} = -V_{SBmax}$. The GPD is fitted to the right tail of the $\hat{V}_{SBmax}$ distribution. The tail modeling procedure is as follows:

1. We performed 5000 MC simulations on an SRAM cell to obtain the histogram of V_{SBmax} (see Figure 4.1). At each MC iteration, the source-bias voltage (V_{SB}) of the cell under test is raised from $0V$ by an increment of $\Delta = 10mV$ until the cell fails. At each voltage, two transient simulations, one with initial data '1' and the other with '0', are performed. In the transient simulations, the cell is switched to the sleep mode by raising its source-bias voltage to the current V_{SB}, and is kept in that mode for a sufficient period, e.g., $2ms$ [54]. Then, the cell is awakened and its data is read out. At some applied V_{SB}, the cell fails, i.e., flips when awakened, in a number of MC iterations. The histogram of V_{SBmax} is obtained by dividing the number of accumulating failed iterations at each V_{SB} by the total number of iterations, i.e., 5000.

2. The data from these MC simulations are used to train the SVM classifier in the SB tool set. We used the 97-th percentile points of $\hat{V}_{SBmax}$ distribution as the classification threshold (t_c). A classifying threshold smaller than the tail threshold is used to avoid the misclassification of the true tail points [47].

3. We used MATLAB to generate 100,000 points in the statistical parameter space, where each point is a vector of 6 Gaussian random variables representing the threshold voltages of transistors in an SRAM cell. Then, we ran the classifier to identify the points which could possibly belong to the tail of the distribution. For example, the classifier returned 10759 points, when $\sigma Vt/Vt = 10\%$.

4. We performed MC simulations on the SRAM cell under test using these prospective tail points and detected V_{SBmax} at each point with a procedure similar to step 1. We calculated the tail threshold (t) as the 99-th percentile point, i.e., 1000-th worst-case V_{SBmax} for 100,000 points. Then, we identified data points that peaked over t.

5. Finally, a GPD is fitted to the data points that peaked over threshold t. A probability weighted moments (PWM) estimator [47] is used to compute the parameters (ξ, β) of the best GPD fit to the exceedance points. We transformed the probability density function (PDF) of the GPD to $G_{\xi,\beta}(-z+t)$ in order to model the left tail of the V_{SBmax} distribution. (Note that the $-z$ mirrors the GPD distribution and the t shifts it right to the tail threshold.)

For example, the estimated PDF of the fitted GPD to the tail of V_{SBmax} distribution when $\sigma Vt/Vt = 10\%$ is shown in Figure 4.2. The estimated parameters for the fitted GPD are ($\xi = -0.3737$, $\beta = 0.0072$). The tail threshold is estimated as $V_{SB} = 0.78V$.

4.3 Computing Array Yield at Elevated V_{SB}

In the following, the derived V_{SBmax} tail distribution is used to estimate the cell failure probabilities at elevated source-bias voltages. Then, relations are developed to compute the failure probability of a whole array based on that of a single cell. The

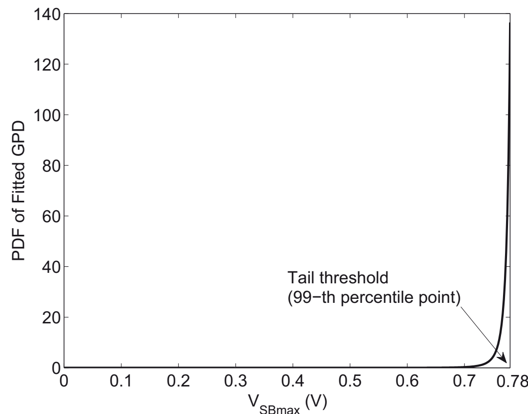


Figure 4.2. Fitted GPD to the tail of V_{SBmax} distribution ($\sigma Vt/Vt = 10\%$).

relation between yield and failure probability of a memory array, i.e., $P_{f,arr}$, is simply

$$Y = 1 - P_{f,arr}. \tag{4.4}$$

4.3.1 Cell Failure Probability at Elevated V_{SB}

Failure probability of an SRAM cell at a given V_{SB} can be defined as

$$\begin{aligned} p_{f,cell}(V_{SB}) &= Pr(V_{SBmax} < V_{SB}) \\ &= F_{V_{SBmax}}. \end{aligned} \tag{4.5}$$

Here, we assume that V_{SBmax} of cells is a random variable with a CDF equal to $F_{V_{SBmax}}$. Using the GPD model for the V_{SBmax} distribution, cell failure probability at elevated source-bias voltages, i.e., $p_{f,cell}(V_{SB})$, can be calculated using (4.5). Figure 4.3 shows the cell failure probability up to the first percentile point of V_{SBmax} distribution, i.e., $V_{SB} = 0.78V$, when $\sigma Vt/Vt = 10\%$. The CDF of the GPD is scaled by a factor of 0.01, as the tail threshold is chosen as the first percentile point. This simple closed-form GPD model allows us to make predictions far out in the tail without having actual simulation data in those regions. For example, the cell failure

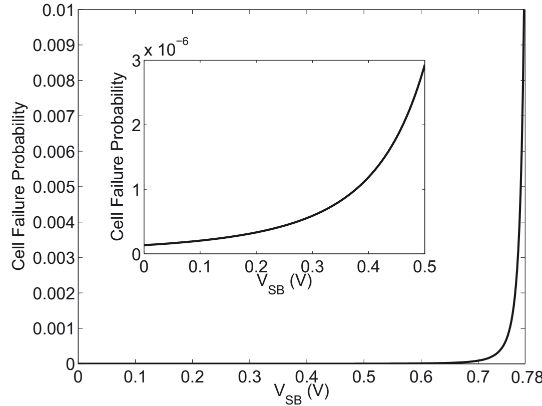


Figure 4.3. Cell failure probability at elevated source-bias voltages up to the first percentile point. ($\sigma Vt/Vt = 10\%$).

probability at $V_{SB} = 0V$, i.e., the normal operating mode, is predicted as $\sim 10^{-7}$, assuming 10% random variation in Vt.

4.3.2 Array Failure Probability at Elevated V_{SB}

The failure probability of a memory array at elevated source-bias voltages depends on its organization. Here, without loss of generality, we choose a direct-mapped cache architecture and investigate its failure probability during the sleep mode. Figure 4.4 shows the block diagram of the investigated cache macro [62]. The cache is divided into two arrays: data and tag arrays. To prevent the access latencies, the tag array is not switched to the sleep mode [22, 19]. Hence, it is only the data array that becomes vulnerable to DRFs during the sleep mode. A data array is usually organized as an array of $M \times w$ data blocks as shown in Figure 4.4, where M is the number of rows and w is the number of data blocks per row. The size of a data block (n) is the number of data bits that are read out at every access of the memory, e.g., 8, 16, 32 bytes. If ECCs are incorporated in the design of the memory, then r check bits are associated with each data block as shown in Figure 4.4.

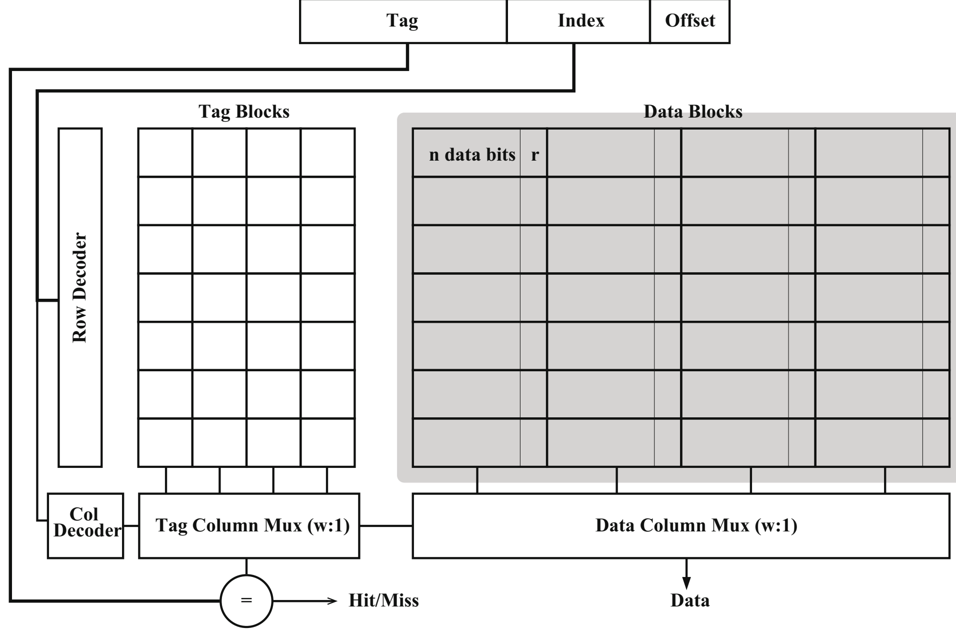


Figure 4.4. Organization of a typical on-chip cache. Leakage reduction is applied only to the data array.

4.3.2.1 Non Fault Tolerance

A memory without a repair mechanism will be non-functional even if one of its cells fails at an elevated source-bias voltage. Hence, V_{SB} applied to an SRAM array should be less than or equal to V_{SBmax} of all cells within the array. Thus, array failure probability at elevated V_{SB} can be written as

$$\begin{aligned} &P_{f,arr}(V_{SB}) = \\ &1 - Pr(V_{SBmax_1} \geq V_{SB}, ..., V_{SBmax_N} \geq V_{SB}) \end{aligned} \tag{4.6}$$

where V_{SBmax_i} is the maximum source-bias voltage of the ith cell in the array and $N = M \times w \times n$ is the total number of cells. Assuming that $\{V_{SBmax_i}, i = 1 : N\}$ are N independent and identically-distributed ($i.i.d.$) random variables, we have

$$\begin{aligned} P_{f,arr}(V_{SB}) &= 1 - \big(1 - Pr(V_{SBmax} < V_{SB})\big)^N \\ &= 1 - \big(1 - p_{f,cell}(V_{SB})\big)^N. \end{aligned} \tag{4.7}$$

4.3.2.2 Redundancy

Assuming a row-redundancy repair scheme, a memory array with M rows and R redundant rows is repairable if the number of failing rows is less than or equal to R. Hence,

$$P_{f,arr}(V_{SB}) = \\ 1 - \sum_{i=0}^{R} \binom{M}{i} p_{f,row}(V_{SB})^i \big(1 - p_{f,row}(V_{SB})\big)^{M-i} \tag{4.8}$$

where, $p_{f,row}(V_{SB})$ is the failure probability of a row when its source-bias voltage is raised to V_{SB}. A row contains $n \times w$ cells, and it fails if any of its cells fail. Hence,

$$P_{f,row}(V_{SB}) = 1 - \big(1 - p_{f,cell}(V_{SB})\big)^{n \times w}. \tag{4.9}$$

By substituting (4.9) in (4.8), the failure probability of a memory array with a predefined number of redundant rows can be calculated at various source-bias voltages. It should be noted however that for simplicity, we have ignored the possibility of DRFs in the redundant rows.

4.3.2.3 ECC

In an ECC-protected memory, r check bits are added to each data block. The probability of having a data block with i faulty cells at raised V_{SB} can be expressed as

$$P^i_{f,block}(V_{SB}) = \\ \binom{n+r}{i} p_{f,cell}(V_{SB})^i \big(1 - p_{f,cell}(V_{SB})\big)^{n+r-i}. \tag{4.10}$$

ECCs can correct a faulty data block as long as the number of its faulty cells is less than or equal to the correcting capacity, i.e., c, of the deployed error correcting code. However, a row is uncorrectable if $i > c$. Hence, at a given V_{SB}, the probability of

having an uncorrectable data block is

$$P_{uc,block}(V_{SB}) = 1 - \sum_{i=0}^{c} P^{i}_{f,block}(V_{SB}). \tag{4.11}$$

A memory array with $M \times w$ data blocks will fail, even if one of its data blocks is uncorrectable. Thus

$$P_{f,arr}(V_{SB}) = 1 - \big(1 - p_{uc,block}(V_{SB})\big)^{M \times w}. \tag{4.12}$$

Using (4.10), (4.11), and (4.12), failure probability of an ECC-protected memory array can be calculated at elevated source-bias voltages.

4.3.2.4 ECC with Redundancy

ECCs and redundancy resources can be used in a synergistic way to improve yield of memory arrays [63, 45]. In this configuration, rows containing data blocks that cannot be corrected by the deployed ECC are replaced by the spare rows. The probability of having an uncorrectable row is

$$P_{uc,row}(V_{SB}) = 1 - \big(1 - p_{uc,block}(V_{SB})\big)^{w}. \tag{4.13}$$

However, a memory chip can be repaired at a raised V_{SB}, as long as the number of uncorrectable rows does not exceed the number of spare rows. Hence

$$\begin{aligned} &P_{f,arr}(V_{SB}) = \\ &1 - \sum_{i=0}^{R} \binom{M}{i} p_{uc,row}(V_{SB})^{i} \big(1 - p_{uc,row}(V_{SB})\big)^{M-i} \end{aligned} \tag{4.14}$$

where R is the number of spare rows. Substituting (4.10) in (4.13) and then in (4.14), yields the failure probability of an array as a function of the source-bias voltage when the combination of ECC and redundancy are used for fault tolerance.

4.4 Estimating Net Power Savings

4.4.1 Overhead Power Associated with Fault-Tolerance Techniques

Employing fault-tolerance techniques in SRAMs allows higher source-bias voltages to be applied to the memory, and thus larger leakage reductions are obtained. However, these techniques may also incur extra dynamic/leakage power consumption due to their additional circuits, wiping out their leakage reductions. There is a leakage associated with redundant rows/columns, however no extra dynamic power is incurred. For ECC-protected memories, however, in addition to the extra leakage power consumed by the check bits, dynamic power is dissipated during the encoding and decoding of data words. Hence, to evaluate the net power savings of the source-biasing technique, the extra power associated with redundant resources and ECC circuits and check bits needs to be taken into account.

The extra dynamic power dissipated by ECC encoder/decoder circuits is merely consumed during the active periods. During a typical operating period T, a memory is active only for a fraction of the time (T_A) dissipating dynamic power, while for the rest of the time memory is in idle mode and dissipates only leakage power. Hence, for an accurate estimation of power savings by the source-biasing technique in the presence of ECC, the duty cycle of a memory, i.e., T_A/T, needs to be known. Duty cycle of memories can vary extensively across applications. For simplicity, we will consider only the asymptotic case of $T_A/T \rightarrow 0$ to estimate the upper bound of power savings. Thus, the net power savings of the source-biasing technique is calculated as its net leakage reductions. The extra leakage power of redundant rows and check bits, however, is taken into account in the leakage reduction estimations.

4.4.2 Estimating Leakage Power Considering Process Variations

The total leakage power of a memory array is the sum of the leakage powers of its constituent cells. Due to the process parameters variations leakage power consumption of the cells differ within an array. Assuming only random variations in device parameters, the leakage power of the cells are N i.i.d. random variables, where N is the total number of cells in the array. As N is large, by the law of large numbers, the total leakage power of the array can be approximated by multiplying the expected value of the cells' leakage power by the total number of the cells. The expected value of the cells leakage power increases with process variations, and can be estimated by performing a MC simulation and calculating the mean of the leakage distribution. Thus, to estimate the leakage power of a whole array as a function of the source-bias voltage (V_{SB}), we sweep V_{SB} by increments of $\Delta = 10mV$, and at each applied V_{SB} a MC simulation is performed. The means of the leakage distributions resulted from MC simulations are calculated and then are multiplied by the total number of cells.

4.5 Simulation Results and Discussion

In the following, we investigate the yield-leakage tradeoffs in a 64KB SRAM array. The organization parameters of the investigated 64KB memory are shown in Table 4.1. Cell failure probability versus source-bias voltage is estimated as the CDF of the GPD fitted to the tail of V_{SBmax} data obtained by the simulation method described in Section 4.2. Note that only the tail of the cell failure probability distribution is of interest, as the array yield drops to zero atV_{SB}s below the tail threshold. Array failure probability relations developed in Section 4.3 are used to compute the yield of the 64KB memory array as a function of the source-bias voltage.

Table 4.1. The organization of the simulated memory array.

Memory size=64KB SRAM
Number of bits per data block (n) =256 bit (32 byte) Number of rows (M)= 512 Number of blocks per row (w)=4 Number of bits per row ($n \times w$) =1024 bit (4*32 byte)
Number of redundant rows (R)=8, 16, 32
SEC-DED code: Number of correctable bits in a block (c)=1 Number of check bits per block (r)=10
DEC-TED code: Number of correctable bits in a block (c)=2 Number of check bits per block (r)=19

4.5.1 Yield Degradations due to Source-Biasing

Figure 4.5(a) shows the yield of the 64KB memory versus V_{SB}, when no fault tolerance technique is used. Different process variation levels are considered. As can be seen, at identical source-bias voltages, yield losses grow rapidly as the process variations deteriorate. For example, at $V_{SB} = 0.5V$, yield losses are $\sim$ 4% when $\sigma Vt/Vt = 5\%$. However, at the same source-bias voltage, the yield losses rise to $\sim$ 80% when the deviation in the threshold voltage of SRAMs increases to 10%. For a memory with 15% and 20% variation, the yield drops to zero when its source-line voltage is raised to $V_{SB} = 0.5V$. As can be seen, at high variation levels, some dies fail even in the active mode, i.e., when $V_{SB} = 0V$, reducing yield below 100%. Hence, fault tolerance techniques are imperative for low power SRAMs when the variations in process parameters are significant.

Figure 4.5(b) shows the yield of the 64KB memory versus source-bias voltage when different levels of redundancy are used. Variation in threshold voltage of transistors is

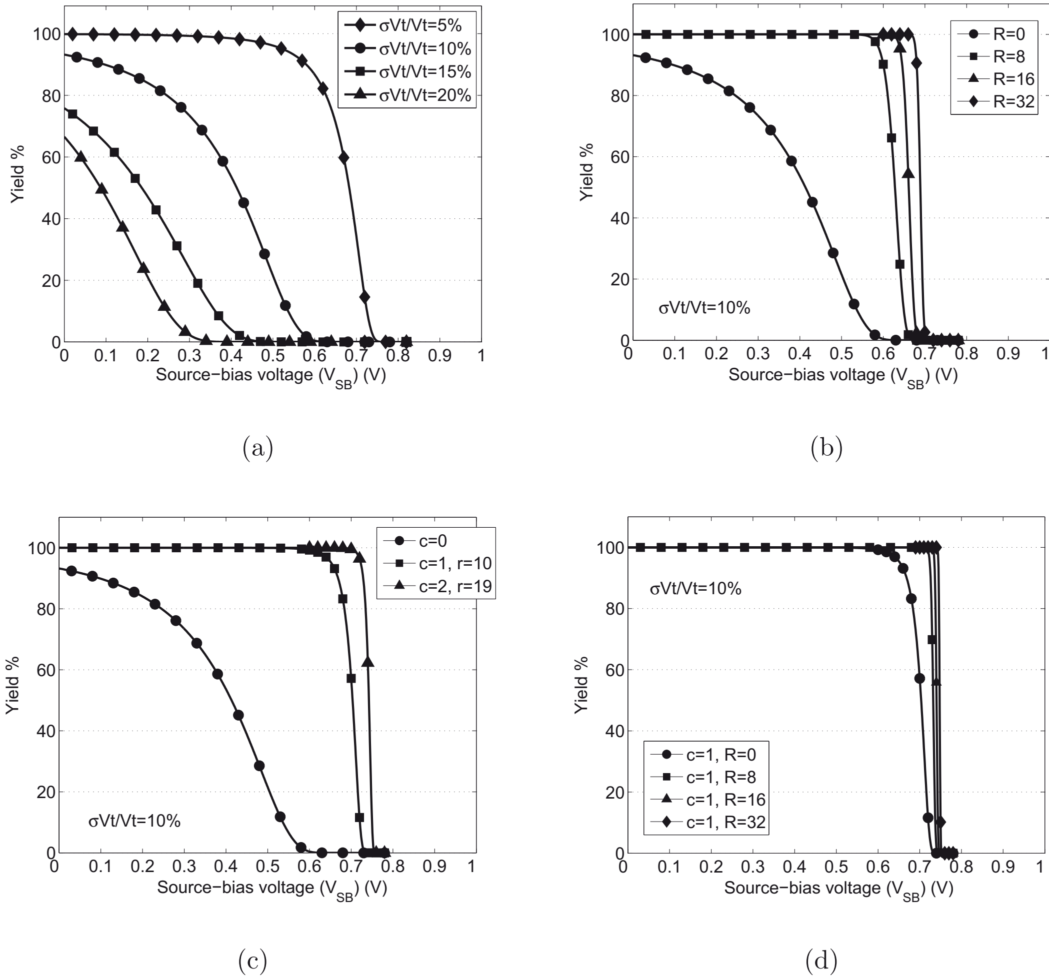


Figure 4.5. Yield of the 64KB SRAM as a function of the source-bias voltage (a) at different levels of process variations when no fault-tolerance technique is present, (b) when $R = 8$, $R = 16$, or $R = 32$ redundant rows are added, (c) when SEC-DED or DEC-TED codes are employed, (d) when $R = 8$ redundant rows in combination with a SEC-DED code are employed.

assumed to be $\sigma Vt/Vt = 10\%$. As can be seen, at this variation level, data-retention failures result in $\sim 10\%$ yield loss even at $V_{SB} = 0V$. Raising V_{SB} beyond $0V$ causes the yield to degrade rapidly, approaching zero at around $V_{SB} = 0.6V$. The yield losses, however, can be avoided by adding a small number of redundant rows, i.e., $R = 8$, as shown in Figure 4.5(b). This eliminates the yield losses due to the DRFs all together at elevated source-line voltages as high as $V_{SB} = 0.5V$. Adding more redundancy, i.e., $R = 16$ or 32, allows only for a marginal increase in the source-bias

voltage. These results indicate that adding a small amount of redundancy is very beneficial in low-leakage SRAMs. However, increasing the available redundancy only offers marginal benefits.

Yield of the 64KB memory versus source-bias voltage when ECCs are employed is shown Figure 4.5(c). Two different ECCs are considered: i) single error correcting double error detecting (SEC-DED) Hamming code (c=1) [64], ii) and double error correcting triple error detecting (DEC-TED) BCH code (c=2) [64]. For the SEC-DED code, the number of check bits (r) added to each 256-bits data block is 10 bits [64]. Whereas for the DEC-TED code, 19 check bits are required [64]. As can be seen, ECCs can considerably enhance the yield due to the random scattering of parametric failures across the memory array. The source-bias voltage of the memory can be raised up to $V_{SB} = 0.5V$ and $V_{SB} = 0.7V$ without any yield losses, using the SEC-DED and DEC-TED codes, respectively. As expected, using a stronger code allows for more aggressive leakage reductions. However, the extra leakage reductions are not significant enough to justify the large area overhead ($\sim 7\%$) of the DEC-TED codes The area overhead of the Hamming SEC-DED code in the investigated 64KB memory stays below $\sim 4\%$.

The yield of the 64KB memory versus source-bias voltage when ECC and redundancy are combined is shown in Figure 4.5(d). Only SEC-DED code, i.e., $c = 1$, is considered in the analysis. The number of available redundant rows are assumed to be $R = 8, 16, 32$. As can be seen, combining redundancy and ECC allows the source-bias voltage to be raised more aggressively pushing the leakage reduction to its fundamental bounds. For example, the source-bias voltage of the 64KB memory can be raised to $V_{SB} = 0.7V$ when a SEC-DED code in combination with $R = 8$ redundant rows are used. However, as can be seen from Figure 4.5(d), increasing the level of redundancy beyond $R = 8$ has only a negligible impact on the yield. Beyond $V_{SB} = 0.7V$, the number of faults increases so sharply that the fault-tolerance techniques are no longer

capable of salvaging the dies. Thus, for ultra low-leakage applications, a combination of a simple ECC and a small number of redundant resources is the most optimal approach to prevent the yield losses due to DRFs.

4.5.2 Yield-Leakage Tradeoff Using Different Fault-Tolerance Techniques

In this section, we investigate the yield-leakage tradeoff governed by the source-bias voltage when different fault-tolerance techniques are used.

4.5.2.1 Yield-Leakage Tradeoffs

The tradeoff between leakage reduction and yield of SRAMs is clearly demonstrated in Figure 4.6. In this figure, the leakage power reductions and the yield of the 64KB memory as a function of the source-bias voltage are shown on the right and left axis, respectively. Different fault-tolerant configurations are considered. Incorporating a SEC-DED code or 8 spare rows in the memory both have a comparable leakage overhead, i.e., $\sim 2\% - 4\%$. When a combination of these techniques are used, their leakage overhead adds up. Hence, two different curves for leakage reductions, each corresponding to one of the above cases, are plotted in Figure 4.6. The relative standard deviation of the threshold voltage variation is assumed to be 10%. As can be seen, raising the source-bias voltage reduces the leakage power, however, it also results in larger yield losses due to the DRFs.

As can be seen from Figure 4.6, without a fault tolerance technique, yield losses can be significant even when the source-bias voltage is raised moderately. For example, raising the source-bias voltage to $V_{SB} = 0.3V$ cuts down leakage by $\sim 70\%$, however it also causes the yield to degrade by $\sim 30\%$. In contrary, using either redundancy

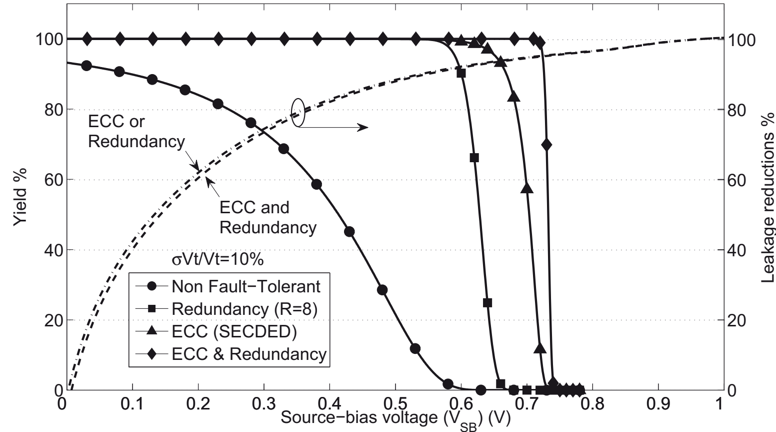


Figure 4.6. The yield-leakage tradeoff in SRAMs: the leakage reductions and yield losses in a 64KB memory as the source-bias voltage is raised.

or ECC allows for $\sim$ 90% leakage reductions, while avoiding any yield losses due to the DRFs during the sleep mode. The combined use of ECC and redundancy, on the other hand, allows for further raising of the source-bias voltage to $V_{SB} \sim 0.7V$, with no entailing yield losses. This configuration allows for reaching the ultimate bounds of leakage reductions, i.e., $\sim$ 95%.

4.5.2.2 Leakage Reductions Subject to a Target Yield

The yield-leakage tradeoff in SRAMs indicates that the source-bias voltage to be applied to a memory design should be determined by considering both the leakage and yield constraints at the same time. For example, subject to a 99% target yield, the limits of the feasible leakage reductions for the 64KB memory using different fault-tolerance techniques are shown in Figure 4.7. The leakage of the memory at each configuration is expressed as a fraction of the raw leakage. Different levels of threshold voltage variations are considered. As can be seen, at low variations, i.e., $\sigma Vt/Vt = 5\%$, the leakage can be reduced down to 30% without using a fault-tolerance technique. Adding $R = 8$ redundant rows, allows for reducing the leakage to $\sim$ 60%. Incorporating ECC alone or combined with redundancy allows for a small

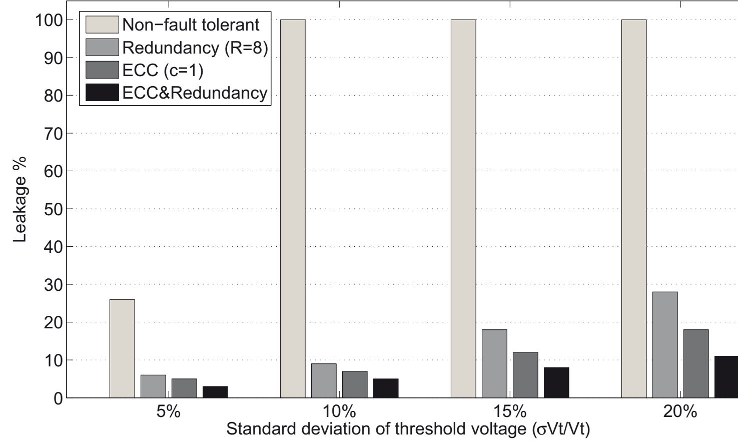


Figure 4.7. Feasible reduced leakage of a 64KB memory using various fault-tolerance techniques subject to a 99% target yield. (100% leakage means no reduction is possible.)

extra reduction of the leakage to $\sim 3\% - 5\%$, at this low variation level. However, as shown in Figure 4.7, subject to the 99% target yield, no leakage reductions can be obtained at high levels of variation when no fault-tolerance technique is used. Adding just a small number of redundant rows, however, allows for significant leakage power reductions with negligible yield losses, i.e., 1%. ECC outperforms the redundancy technique, especially in high variations. However, considering the area and dynamic power overhead of on-line error detection and correction by ECCs, they become less attractive compared to the off-line memory repair using redundant resources. For ultra-low leakage applications, the ultimate bounds of leakage reduction can be obtained by the combined use of ECCs and redundancy. In particular, this approach can be attractive when variations are large and the activity factor of memory is small, so that the overhead associated with the dynamic power of ECC encoding/decoding becomes negligible.

4.6 Summary

Scaling the rail-to-rail voltage of SRAM cells to reduce their leakage power consumption during idle periods also results in the degradation of the cells' robustness, making them vulnerable to data-retention failures (DRFs). Therefore, the main goal while attempting to reduce the leakage power of SRAMs is to limit the yield losses due to the DRFs. In this work, we developed analytical models to investigate the involved yield-leakage tradeoffs in SRAMs. The results show that switching SRAMs to a sleep mode can result in significant yield losses in large arrays due to the parametric DRFs, especially in processes with highly fluctuating parameters. Thus, we investigated the application of fault-tolerance techniques for a more efficient leakage reduction of SRAMs, by providing tolerance to the failures that might occur during the sleep mode. The results show that in a 45-nm technology, assuming 10% variation in the transistors' threshold voltage, repairing a 64KB memory using only 8 redundant rows or incorporating single error correcting codes allows for $\sim 90\%$ leakage reduction while incurring only $\sim 1\%$ yield loss. The combination of redundancy and ECC, however, allows us to reach the ultimate bounds of the leakage reduction, i.e., $\sim 95\%$.

Chapter 5

Post-Silicon Tuning of Standby Supply Voltage in SRAMs to Reduce Parametric Data-Retention Failures

Intra-die variations in process parameters result in a within-die distribution of cells' data-retention voltage (DRV). Hence, the minimum applicable standby voltage to a memory die (V_{DDLmin}) is determined by the maximum DRV among its constituent cells. On the other hand, inter-die variations result in a die-to-die variation of V_{DDLmin}. Applying an identical standby voltage to all dies, regardless of their corresponding V_{DDLmin}, can result in the failure of some dies, due to data-retention failures (DRFs), entailing yield losses. In this chapter, we propose a post-silicon standby voltage tuning scheme to avoid the yield losses due to the DRFs, while reducing the leakage currents effectively.

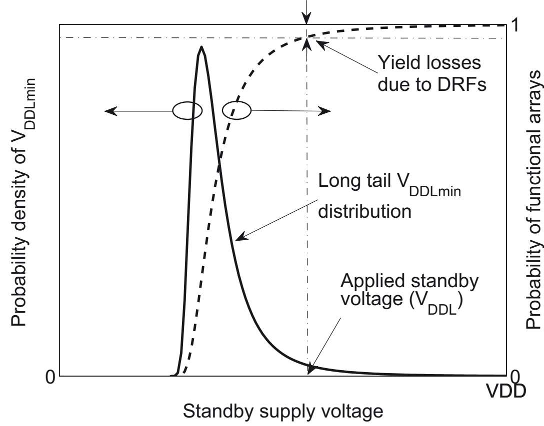


Figure 5.1. A conceptual illustration of the long tail V_{DDLmin} distribution and probability of functional array versus standby supply voltage.

5.1 Inter-Die Distribution of V_{DDLmin}

The minimum applicable standby supply voltage to an SRAM array (V_{DDLmin}) is determined by the worst-case DRV, i.e., the maximum DRV among the array's constituent cells. Systematic inter-die variations in process parameters result in a die-to-die variation of V_{DDLmin}. Hence, if, for example, M memory dies, each containing N, are manufactured and their V_{DDLmin} is detected, the results will exhibit an inter-die distribution. From a mathematical point of view, this is analogous to generating M data blocks each containing N samples from the same distribution, i.e., the underlying DRV distribution, and detecting the maximum values from these M data blocks. *Extreme Value Theory* [55, 65] states that the inter-die distribution of V_{DDLmin} will converge to a *generalized extreme value* (GEV) distribution with a long tail, as conceptually illustrated in Figure 5.1.

Now, let us assume that an identical standby voltage (V_{DDL}) is applied to all these dies, regardless of their corresponding V_{DDLmin}. Then, due to the long tail distribution of V_{DDLmin}, there is always the possibility that some dies might have a V_{DDLmin} above the applied V_{DDL}. Such dies will contain cell(s) with a DRV larger than the applied V_{DDL}. Hence, they will lose their data during the standby mode,

and result in the failure of the whole array due to containing data-retention (hold) failures (DRFs) [41, 34]. The probability of having a functional array at a given V_{DDL}, calculated as the cumulative distribution function (CDF) of V_{DDLmin}, is also shown in Figure 5.1. As can be seen, the probability of array failures due to DRFs starts to increase as soon as the standby voltage is reduced below the nominal V_{DD}, resulting in yield degradations. The size of the yield losses at moderate standby voltage reductions is determined by the tail behavior of the V_{DDLmin} distribution, which in turn is controlled by the underlying DRV distribution.

Post-silicon tuning techniques have been introduced in the literature [66, 67, 68, 69, 70, 71, 72] that allow chip parameters, e.g, clock frequency, operating voltage, etc., to be adjusted after the die has been manufactured, in order to compensate for the specific inter- and intra-die variations that have occurred on that particular die. In this work, we investigate a similar approach as in [69, 70] to tune the standby supply voltage of each individual die after manufacturing, i.e., post-silicon. This allows the yield losses due to the parametric data-retention failures during the standby mode to be avoided, while effectively reducing the leakage power dissipation.

5.2 Minimum Applicable Standby Voltage to a Memory Die (V_{DDLmin})

The existence of even a single DRF in a memory array can result in the failure of the whole array. Hence, the minimum applicable standby voltage to a memory die is determined by the highest DRV among its cells, i.e.,

$$V_{DDLmin} = max(DRV_1, DRV_2, ..., DRV_N) \tag{5.1}$$

where N is the total number of bitcells and $\{DRV_i, i = 1 : N\}$ are the corresponding DRVs of the cells.

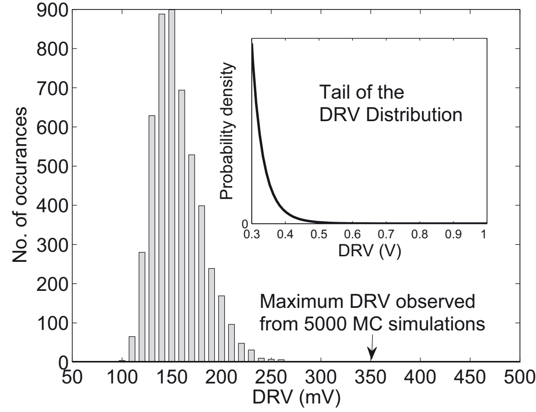


Figure 5.2. Histogram of the DRV from a 5000 point Monte Carlo simulation of SRAM cells in a 45-nm predictive technology node.

Monte Carlo simulations, along with the statistics of variations in device parameters, are traditionally used to estimate the statistical intra-die distribution of DRV for small memory arrays, and then V_{DDLmin} is determined as the upper bound of this distribution [41, 43, 73]. For example, a histogram of the DRV of SRAM cells in a 5Kb memory array, obtained by a Monte Carlo (MC) simulation, is shown in Figure 5.2. Based on these results, the minimum applicable standby voltage to such a memory is estimated as $V_{DDLmin} = 350mV$ (the upper bound of its DRV histogram). However, for large memory arrays, the statistically rare events of cells with a DRV extremely deviated from the median of the distribution also need to be taken into account. The probability of these rare events might be very small, nevertheless, it can be magnified by the sheer number of replicated cells in large SRAM arrays [47, 48, 74]. As can be seen from Figure 5.2, the distribution of DRVs seems to have a long and heavy right tail which is not fully represented due to the limited number of MC simulations. The long tail of the distribution implies that for large memories, the probability of DRFs can be non-zero even at relatively large applied standby voltages.

In general, V_{DDLmin} of a memory die is determined by the following factors:

5.2.1 Joint Impact of Inter- and Intra-Die Variations on V_{DDLmin} of SRAMs

V_{DDLmin} of a memory die is determined by the specific inter- and intra-die process parameters variations that have occurred on its transistors. Random intra-die variations result in a within-die statistical distribution of cells DRV. Hence, V_{DDLmin} of a memory increases as the intra-die variations in device parameters aggravates. On the other hand, systematic inter-die variations in process parameters cause the median of the within-die DRV distribution to vary from die to die. For example, the mean of the cells DRV is expected to be larger in dies at the FS (Fast NMOS, Slow PMOS) corner of the process, due to a larger leakage current of pull-down NMOS transistor [72]. As a result, V_{DDLmin} of dies from the FS corner of the process will be larger than that of dies subject to nominal process.

5.2.2 Impact of the Size of Memory on its V_{DDLmin}

Due to the long tail distribution of the DRV, the probability of DRFs is non-zero, nevertheless extremely low, at large standby voltages. However, as the size of a memory array grows, the probability of it containing DRFs is magnified in proportion with the number of its cells. Thus, V_{DDLmin} increases with the size of a memory [42].

5.2.3 Impact of Adding Redundancy on V_{DDLmin} of SRAMs

Adding redundancy to a memory allows the replacing of cells that fail at the applied V_{DDL} with available spares after manufacturing. Thereby, the dies that have a limited number of DRFs can be salvaged. This allows the standby voltage of a memory die to be reduced more aggressively by providing tolerance to some DRFs. However, the standby voltage of a memory can only be reduced down to a point where

the number of DRFs does not exceed the repair capability of the memory. V_{DDLmin} can be further reduced by increasing the number of redundant resources.

5.2.4 Mathematical Model of Inter-Die V_{DDLmin} Distribution

Let us assume that N_{dies} memory dies, each containing N_{cells}, are manufactured and their V_{DDLmin} is detected. From a mathematical point of view, this is analogous to generating N_{dies} data blocks each containing N_{cells} samples from the same distribution, i.e., the underlying global DRV distribution, and taking the maximum values from these N_{dies} data blocks. The distribution of the memory dies V_{DDLmin} can be studied using a branch of statistics, called *Extreme Value Theory* (EVT) that deals with the behavior of the block maxima. The classical EVT states that, if blocks of a large number of independent random values are generated from a single probability distribution F, the maxima of the blocks will converge in distribution to a random variable with a *generalized extreme value* (GEV) distribution [55, 65]. It is shown that the theory is equally applicable to dependent data as long as the long-range dependence at extreme levels is weak [55]. The DRV of adjacent cells are not completely independent and exhibit a small spatial correlation [43], and thus the classical EVT is not directly applicable to the study of the dies V_{DDLmin} distribution. However, it has been shown that the theory is equally valid for dependent data as long as the long-range dependence at extreme levels is weak [55].

Ignoring the location and scale parameters, the CDF of the GEV distribution can be restated as

$$H_\xi(x) = \begin{cases} e^{-(1+\xi x)^{-1/\xi}}, & \xi \neq 0 \\ e^{-e^{-x}}, & \xi = 0, \end{cases} \quad \text{where } 1 + \xi x > 0. \tag{5.2}$$

GEV unites the Gumbel, the Frechet and the Weibull distributions into a single

family [55]. The shape parameter ξ determines the type of the GEV distribution, and thereby its tail behavior:

1. When $\xi > 0$, the GEV is equivalent to the Frechet distribution, which has a lower bound $(-1/\xi)$ and a heavy right tail.
2. When $\xi < 0$, the GEV is equivalent to the Weibull distribution, which has an upper bound $(-1/\xi)$ and a heavy left tail.
3. In the limit, as $\xi \rightarrow 0$, the GEV becomes the Gumbel distribution, which is unbounded.

The original distribution, i.e., F, determines the shape parameter, ξ, of the resulting GEV distribution, and thereby its tail behavior.

5.2.5 Tradeoff between Leakage Reduction and Yield of SRAMs

Scaling the supply voltage is neseccary to reduce yield losses due to the memory dies with a leakage higher than a predefined budget. However, due to a long tail distribution of V_{DDLmin} of dies, the probability of array failures drastically increases as the standby voltages is reduced, causing large yield degradations. This establishes a tradeoff between leakage reduction and yield of SRAMs. In the following, we first describe a simulation methodology to estimate data-retention failure probability of a single SRAM cell at reduced supply voltages. Then, analytical models are developed to compute the yield of a whole array as a function of the standby voltage based on the failure probability of a single cell. We use these results to derive the empirical CDF of V_{DDLmin}, and then a GEV distribution is fitted to this data, using maximum likelihood estimation (MLE) method, to investigate the tail behavior of the distribution.

Then, we investigate the possibility of a more aggressive approach by independently adjusting the V_{DDL} of each individual die to its corresponding V_{DDLmin}, and evaluate its yield enhancement effeciency.

5.3 Estimating Data-Retention Failure Probability as a Function of Supply Voltage

An SRAM cell fails to retain its data if its supply voltage is reduced below its DRV. Hence, the probability of a data-retention failure at a reduced supply voltage V_{DDL} can be written as

$$\begin{aligned} p_{f,cell}(V_{DDL}) &= Pr(DRV > V_{DDL}) \\ &= 1 - Pr(DRV \leq V_{DDL}). \end{aligned} \tag{5.3}$$

Given the statistical distribution of the cells DRV, $p_{f,cell}(V_{DDL})$ can be calculated using (5.3). Simulation or analytical approaches, along with the statistical parameters of device variations, can be used to estimate the distribution of DRV for a given SRAM array with a certain size. Analytical approaches suffer from the approximations that are necessary to make the statistical analysis of DRV tractable [40, 34, 75, 47]. Hence, Monte Carlo (MC) simulations are generally used to obtain DRV distribution of cells within a memory array.

5.3.1 Estimation of Rare Failure Events

In large memory arrays, estimating even rare failure events is crucial, as their probability is magnified by the sheer number of replicated bitcells [47, 48]. For example, in a 1Mb memory, with 1 million replicated bitcells, even a data-retention failure probability as low as 10^{-8} can result in 1% yield loss. The number of MC simulations

required to observe a rare event in the tail of the DRV distribution, i.e., an event resulting in a cell with a DRV extremely deviated from the nominal value, is inversely proportional to the probability of that event. Hence, a huge number of MC simulations are required to estimate a sufficiently accurate statistics on these rare events in large memory arrays. Using a limited number of samples will fail to accurately model the tail of a distribution such as the one shown in Figure 5.2. Extrapolating this distribution in an ad-hoc manner, for example using a normal distribution, can also lead to gross inaccuracies. Thus, to obtain a sufficiently accurate approximation of the tail of the DRV distribution in a reasonable simulation time, we use the mixture importance sampling technique presented in [48]. Cell failure probability at various voltages, i.e., $p_{f,cell}(V_{DDL})$, is later computed using (5.3). The proposed simulation methodology is described in the following section.

5.3.2 Simulation Methodology

We performed simulations to obtain an approximation to the failure probability of SRAM cells as a function of the standby voltage. We first designed an SRAM cell using device models from the 45-nm Predictive Technology Model (PTM) [17]. Minimum feature size transistors are used for pull-up PMOS and access NMOS transistors of the cell. For pull-down NMOS transistors, minimum length is used. However, their width is set so that a cell ratio and write ratio of 2 and 1 are obtained, respectively, i.e., $Wn/Wa = 2$, and $Wp/Wa = 1$, where Wn, Wa, and Wp, are the width of pull-down, access, and pull-up transistors. Then we performed MC simulations combined with the mixture importance sampling technique [48] to obtain an approximation to $p_{f,cell}(V_{DDL})$. All simulatiions are performed at $T = 70°C$. As there is no process variation technology file available for the predictive technology models, we use a

methodology similar to the one presented in [60, 68, 72] to model process variations in our MC simulations. The modeling methodology is described in the following section.

5.3.2.1 Inter- and Intra-Die Process Variation Modeling

Process variations impact various device parameters, e.g., channel length, gate-oxide thickness, threshold voltage etc. For simplicity, we have restricted our model only to a variation in the threshold voltage (Vt) of transistors. Figure 5.3 illustrates the modeling methodology for global and local variations in Vt of an SRAM cell's transistors. Threshold voltages of transistors of a certain SRAM cell are affected by both the global and local process variations. Due to the global variations, i.e., die-to-die, wafer-to-wafer, etc., threshold voltages of all NMOS and PMOS transistors of an SRAM cell are shifted from their nominal value by a certain amount. Whereas, local random variations result in deviation of the threshold voltage of each individual transistor from its shifted-nominal value, i.e., Vt_{inter_i}, resulting in mismatches among them.

The inter-die shifts in Vt of NMOS and PMOS transistors are generally uncorrelated due to their different process steps [7]. Hence, an accurate analysis would require an assumption of a two-dimensional Gaussian distribution for the inter-die threshold voltage variation of NMOS and PMOS transistors. However, such a multi-dimensional analysis can add a lot of computational cost and complexity. Hence, we project the two-dimensional Gaussian distribution of Vt shift for NMOS and PMOS transistors to a one-dimensional Gaussian to reduce the complexity of our analysis. This projection can be realized along different directions in the Vt shift plane. We choose the anti-correlated Vt axis, where the shifts in Vt of NMOS and PMOS transistors are in different directions (see Figure 5.3), as the projection axis.

The above assumption is to create the pessimistic variation scenarios for the DRV

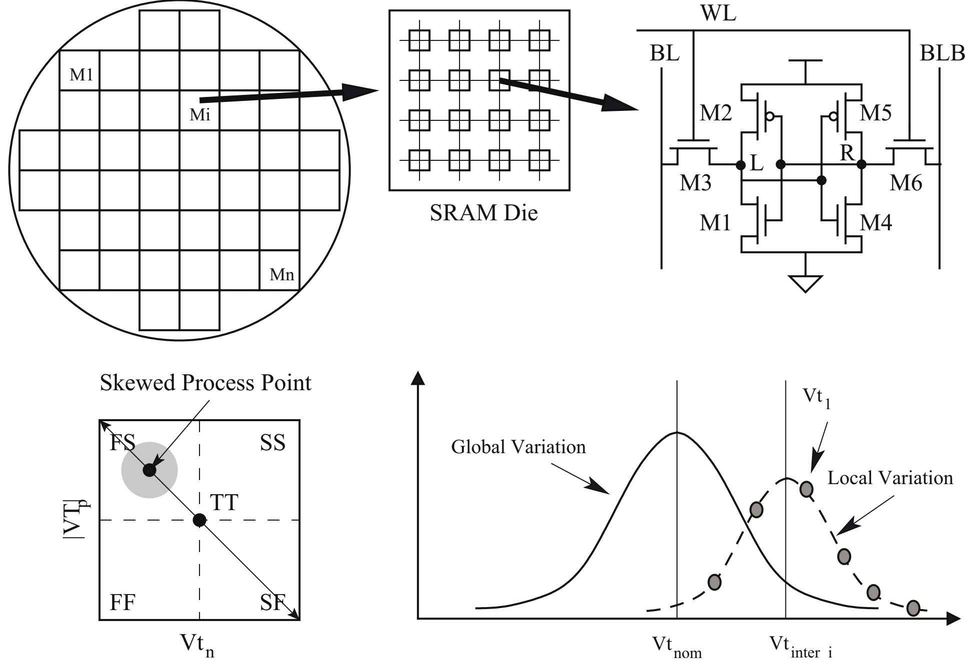


Figure 5.3. Inter and intra-die variation modeling.

of SRAM cells. For example, suppose that the SRAM cell in Figure 5.3 is storing data '1', i.e., $V_L = 1V$ and $V_R = 0V$. DRV of such an SRAM cell will be the largest in the following conditions: i) when M1 is fast and M2 is slow, i.e., FS corner of the process: this condition facilitates the reduction of V_L below the supply voltage due to the higher leakage currents of M1, causing the cell to lose data '1' at higher supply voltages. ii) when M4 is slow and M5 is fast, i.e., SF corner of the process: this condition increases the switching voltage, i.e., the input voltage at which the inverter switches its output, of the M4-M5 inverter, making the cell flip at higher supply voltages. Thus, as shown in Figure 5.3, we model the inter-die variations as an equal shift in Vt of NMOS and PMOS transistors, so that the dies are biased toward either the FS or SF corner of the process.

The local variations in Vt of transistors in the SRAM cell are modeled as six random variables. Thus, as shown in Figure 5.3, we first apply a certain amount of global Vt shift (ΔVt_{inter}) to all six transistors of the cell. Then, on top of this global

variation, a certain amount of local random Vt shift is imposed to each individual transistor to represent mismatches. Hence, the threshold voltage of each transistor is expressed as the sum of the nominal threshold voltage (Vt_{nom}) combined with the inter- and intra-die variations

$$Vt = Vt_{nom} + \Delta Vt_{inter} + \Delta Vt_{intra}. \tag{5.4}$$

Distribution of inter- and intra-die variations in Vt are assumed to be normal with different variances

$$\begin{aligned} \Delta Vt_{inter} &\sim N(0, \sigma_{Vt_{inter}}) \\ \Delta Vt_{intra} &\sim N(0, \sigma_{Vt_{intra}}). \end{aligned} \tag{5.5}$$

5.3.2.2 Importance Sampling

Importance sampling (IS) in Monte Carlo (MC) simulations allows to increase the precision of the estimates that can be obtained by a given number of iterations. The basic idea of IS is to generate samples from a biased distribution rather than the original one, in order to increase the population of the samples from the region of interest in the distribution, e.g., the tail. We use a modified version of the IS, called the mixture importance sampling (MixIS) technique [76, 48], in our MC simulations to estimate the small $p_{f,cell}(V_{DDL})$ within a reasonable simulation time. In this technique, the samples are generated from a distribution which is a mixture of normal (N) and uniform (U) distributions

$$g(X) = \lambda_1 N(X) + \lambda_2 U(X) + (1 - \lambda_1 - \lambda_2) N(X - \mu_s) \tag{5.6}$$

The Uniform distribution in $g(X)$ (the second term in (5.6)) assures that no region is left unsampled, while the shifted original distribution (the third term in (5.6)) enables focusing on the failure regions. The scaled original distribution (the first term in (5.6)) prevents the under-representation of the body of the sample space. We

use MATLAB to generate N_{IS} random vectors X from g. X is a vector of six *i.i.d.* random variables, representing the random intra-die threshold voltage variations of the six transistors in the SRAM cell (see Figure 5.3)

$$X = (\Delta Vt_1, ..., \Delta Vt_6). \tag{5.7}$$

Parameters λ_1, λ_2 and μ_s are determined with the methods presented in [48].

The random numbers generated from the mixture distribution g are passed as threshold voltage variations to the six transistors of the SRAM cell under investigation. Two transient HSPICE simulations, one with initial data '1' and the other with '0', are performed at each MC iteration, where the cell is put in standby mode for an appropriate time, e.g., $2ms$, and then is awakened and its data is read out. The supply voltage is swept from V_{DD} to 0 by a $10mV$ decrement at each MC iteration, and the DRV of the SRAM cell in that particular iteration is determined as the minimum V_{DD} at which the data is still retained, i.e., the cell does not flip when awakened.

The simulation outputs of the IS technique should be appropriately weighted to compensate for the use of a biased sampling distribution. Various estimates can be used to compute the expected value of the output random variable in the IS technique [76]. The classical MC integration estimate [76] can fail as its weights do not sum to 1. Thus, we use a normalized estimate, called the ratio estimate [76], to compute the expected value of the cell failures at a given V_{DDL} using the DRV data from our MC simulations

$$p_{f,cell}(V_{DDL}) = \frac{\sum_{i=1}^{N_{IS}} w(X^i) I(X^i, V_{DDL})}{\sum_{i=1}^{N_{IS}} w(X^i)} \tag{5.8}$$

where

$$w(X) = \frac{N(X)}{g(X)} = \frac{\prod_{i=1}^{6} N(\Delta V t_i)}{\prod_{i=1}^{6} g(\Delta V t_i)} \tag{5.9}$$

and

$$I(X^i, V_{DDL}) = \begin{cases} 0, & \text{cell retains its data at } V_{DDL} \\ 1, & \text{cell loses its data at } V_{DDL}. \end{cases} \tag{5.10}$$

In the above equations, N is the original intra-die distribution of threshold voltage variation, which is assumed to be a normal distribution as expressed in (5.5). $w(X)$ is called the weight function and is equal to the ratio of the original distribution to the distorted one. As the threshold voltages of transistors in a single cell are assumed to be six independent random variables, their joint distribution in (5.9) is factored into the product of their individual probability density functions. We consider as much as 10% relative standard deviation (RSD), i.e., σ/μ, for the intra-die variations in Vt.

To investigate the impact of inter-die variations on the cell failure probability, we repeated the above procedure with Vt of NMOS and PMOS transistors shifted from their nominal value along the axis passing through the FS and SF corners as shown in Figure 5.3. To limit the number of simulations, $\Delta V t_{inter}$ is swept from $-250mV$ to $+250mV$ by large increments of $25mV$. The computed cell failure probabilities ($p_{f,cell}(V_{DDL})$) at sampled skewed process points are then interpolated to obtain a continuous function for the cell failure probability versus Vt shift. This function is later used to compute the overall memory array yield.

The accuracy of the failure probability estimates by the devised importance sampling technique is controlled by the number of simulated samples (N_{IS}). Thus, N_{IS} should be chosen carefully to ensure the accuracy of estimations while keeping the simulation-time tractable. Authors in [48] report that the estimates for failure probabilities as low as 10^{-9}, with 95% confidence interval equal to $\pm 10\%$ error range,

converge to their real value with only 2000-3000 MixIS samples in a statistical SRAM stability analysis example, achieving a speedup of $\sim 100\times$ compared to a regular MC simulation. Moreover, they show that despite the regular MC simulation, the number of MixIS simulations does not increase as the target failure probability decreases. In this work, the reported values for the cell failure probability are obtained by $N_{IS} = 5000$ simulations.

5.4 Computing Array Yield from Cell Failure Probability

In the following, we develop mathematical relations to allow us to compute the yield of a complete memory array from the failure probability of one constituent cell. Here, we talk in terms of failure probability rather than yield to avoid the use of confusing negatives. The relation between yield and failure probability of a memory array, i.e., $P_{f,arr}$, is simply

$$Y = 1 - P_{f,arr}. \tag{5.11}$$

5.4.1 Yield of a Memory Without Redundancy

If the applied standby voltage to a memory array, i.e., V_{DDL}, is smaller than its corresponding V_{DDLmin}, then at least one of its cells will fail. This can result in the failure of the whole array, and cause yield degradation when there is no repair mechanism. Thus, array failure probability at reduced standby voltage V_{DDL} can be written as

$$\begin{aligned} P_{f,arr}(V_{DDL}) &= Pr(V_{DDLmin} > V_{DDL}) \\ &= 1 - Pr(V_{DDLmin} \leq V_{DDL}). \end{aligned} \tag{5.12}$$

$Pr(V_{DDLmin} \leq V_{DDL})$ is the probability that all N cells in the array have a DRV below the applied V_{DDL}. Thus we can write

$$\begin{aligned} &Pr(V_{DDLmin} \leq V_{DDL}) = \\ &Pr(DRV_1 \leq V_{DDL}, ..., DRV_N \leq V_{DDL}). \end{aligned} \tag{5.13}$$

Assuming that $\{DRV_i, i = 1 : N\}$ are N *i.i.d.* random variables, we have

$$Pr(V_{DDLmin} \leq V_{DDL}) = Pr(DRV \leq V_{DDL})^N. \tag{5.14}$$

Substituting (5.3) in (5.14) and (5.12) yields

$$P_{f,arr}(V_{DDL}) = 1 - (1 - p_{f,cell}(V_{DDL}))^N. \tag{5.15}$$

5.4.2 Yield of a Memory With Redundancy

A memory with redundancy can be repaired if the number of failures is limited. Assuming a row-redundancy scheme, a memory array with M rows and R redundant rows is not repairable if the number of failing rows is more than the number of redundant rows. Hence,

$$\begin{aligned} &P_{f,arr}(V_{DDL}) = \\ &1 - \sum_{k=0}^{R} \binom{M}{k} p_{f,row}(V_{DDL})^k \left(1 - p_{f,row}(V_{DDL})\right)^{M-k} \end{aligned} \tag{5.16}$$

where $p_{f,row}(V_{DDL})$ is the failure probability of a row at V_{DDL}. A row with N/M cells fails if any of its cells fail. Hence,

$$P_{f,row}(V_{DDL}) = 1 - (1 - p_{f,cell}(V_{DDL}))^{N/M}. \tag{5.17}$$

Using (5.17) and (5.16), the failure probability of a memory array with a certain number of redundant rows can be calculated at various standby voltages.

It should be noted that, in the above analysis, we have assumed that DRFs do not occur in the redundant rows. However, it is clear that the process parameters

variations can also impact redundant cells and result in the deviation of their DRV as well. The above assumption may be justified though, as the redundant cells can be made more robust against variations by trading off area for better data-retention capability. Larger transistors can be used in SRAM cells of redundant rows, for example, to make them retain their data more reliably. Moreover, larger transistors are less affected by process parameters variations [7], diminishing the deviations of the DRV of the redundant cells. Redundant cells occupy a small area of a memory die, e.g., 1%, thus the impact of increasing their size on the area of the whole array is relatively small and can be neglected in first approximation.

5.4.3 Poisson Yield Model

Equations (5.15) and (5.16) can be hard to compute directly, because $\binom{M}{k}$ can be inconveniently large and $p_{f,row}(V_{DDL})^k$ can be inconveniently small. A more convenient relation can be obtained using a Poisson approximation to the binomial distribution in (5.15) and (5.16).

We can define the average number of failing cells at V_{DDL} in a memory array as

$$\lambda(V_{DDL}) = N \times p_{f,cell}(V_{DDL}) \tag{5.18}$$

where N is the total number of bitcells in the array. Then, (5.15) can be approximated as

$$\begin{aligned} P_{f,arr}(V_{DDL}) &= 1 - \left(1 - \frac{\lambda(V_{DDL})}{N}\right)^N \\ &\approx 1 - e^{-\lambda(V_{DDL})}, \end{aligned} \tag{5.19}$$

when N becomes sufficiently large.

Similarly, (5.17) can be approximated as

$$
\begin{aligned}
P_{f,row}(V_{DDL}) &= 1 - \left(1 - \frac{\lambda(V_{DDL})}{N}\right)^{N/M} \\
&\approx 1 - e^{-\lambda(V_{DDL})/M}.
\end{aligned}
\tag{5.20}
$$

The number of rows (M) is large enough in typical memory arrays to make the binomial distribution in (5.16) approach a Poisson distribution

$$
P_{f,arr}(V_{DDL}) = 1 - \sum_{k=0}^{R} \frac{\lambda_{row}(V_{DDL})^k e^{-\lambda_{row}(V_{DDL})}}{k!} \tag{5.21}
$$

where λ_{row} is the average number of faulty rows at V_{DDL}, i.e.,

$$
\lambda_{row}(V_{DDL}) = M \times p_{f,row} = M\left(1 - e^{-\lambda(V_{DDL})/M}\right). \tag{5.22}
$$

From estimates of the failure probability of an SRAM cell at various V_{DDL}, we can compute the yield of a complete memory array as a function of the standby voltage with and without redundancy, using (5.19) and (5.21), respectively.

5.5 Applying an Identical Standby Voltage to All Dies

Probability of data-retention failures at a certain standby voltage varies from die-to-die due to inter-die variations in process parameters. This implies that, if an identical standby voltage is applied to all dies regardless of their actual process parameters, the number of DRFs may become too large in some dies with strongly skewed process parameters, making them non-repairable by the available redundancy resources, and thereby drastically impacting the overall yield. In the following, we investigate the relationship between the yield losses and the applied standby voltage in SRAMs.

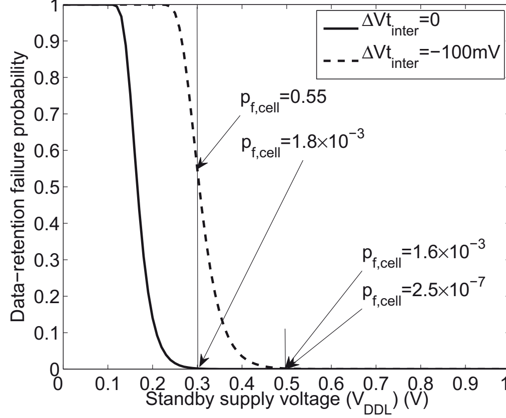


Figure 5.4. (a) Probability of DRFs versus standby voltage at two different inter-die corners.

5.5.1 Impact of Inter-die Variations on Data-Retention Failure Probability

Inter-die variations impact both the leakage and failure probability of SRAMs. Probability of data-retention failures versus standby voltage at the nominal ($\Delta Vt_{inter} = 0mV$) and the FS corner of the process ($\Delta Vt_{inter} = -100mV$) are shown in Figure 5.4. Cell failure probabilities are obtained using the importance sampling technique described in Subsection 5.3.2. As can be seen, at identical standby voltages, the probability of DRFs is higher in the memory die at the FS corner of the process compared to the die at the nominal corner. For example, if the standby voltage is reduced to $V_{DDL} = 0.5V$ for both of these memories, failure probability will be 2.5×10^{-7} for the die at $\Delta Vt_{inter} = 0$, while it is 1.6×10^{-3} for the die at $\Delta Vt_{inter} = -100mV$. If the standby voltage is reduced more aggressively to $V_{DDL} = 0.3V$, the probability of data-retention failures increases for dies at both process points. However, the die at the FS corner is impacted more severely, where in average more than half of its cells fail. Note that the probability of a DRF only asymptotically approaches zero as the standby voltage is increased. For example, at $V_{DDL} = 1.0V$, i.e., no supply voltage reduction (active mode), cell failure proba-

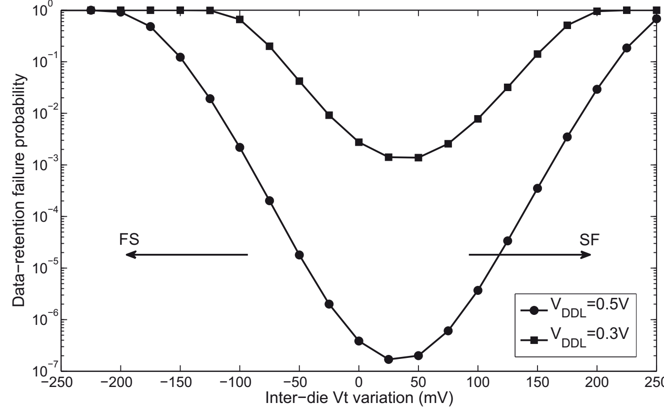


Figure 5.5. Probability of DRFs at various inter-die corners for $V_{DDL} = 0.5V$ and $V_{DDL} = 0.3V$.

bility becomes extremely low but non-zero, namely 1.1×10^{-16} and 3.0×10^{-10} at $\Delta Vt_{inter} = 0$ and $\Delta Vt_{inter} = -100mV$, respectively.

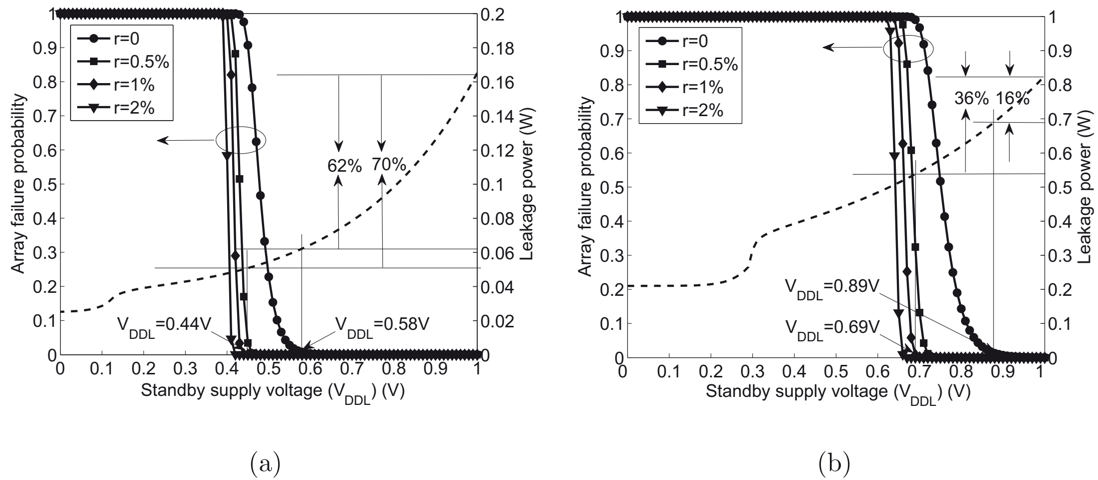
Data-retention failure probability at different skewed process points is shown in Figure 5.5, when standby voltage is reduced to $V_{DDL} = 0.5V$ and $V_{DDL} = 0.3V$. As can be seen, at all process parameters settings, the failure probability grows when the standby voltage is reduced from $V_{DDL} = 0.5V$ to $V_{DDL} = 0.3V$. However, at both standby voltages, the probability of data-retention failures increases when the process is skewed toward the FS or the SF corner of the process. The increase in the probability of DRFs in dies skewed toward the FS corner is due to the larger leakage currents of pull-down transistors in the SRAM cell [34]. Whereas, in dies skewed toward the SF corner, the increase in failure probability is due to the increased switching voltage of the cross-coupled inverters of SRAM cells [34].

5.5.2 Impact of Inter-die Variations on Array Failure Probability

As shown in the previous section, the probability of data-retention failures at a certain standby voltage, i.e., $p_{f,cell}(V_{DDL})$, increases as Vt deviates far away from its nominal value. However, note that the smallest failure probabilities were observed for small positive shifts of the transistor thresholds (see Figure 5.5). This implies that, if an identical standby voltage is applied to all dies regardless of their skewed process parameters settings, the number of DRFs may become too large in some dies with strongly skewed process parameters. This can make some dies non-repairable by the available redundancy resources, and thereby drastically impact the overall yield. Figure 5.6 shows array failure probability of a 1Mb memory die versus standby voltage at nominal Vt (Figure 5.6(a)) and $\Delta Vt_{inter} = -100mV$ (Figure 5.6(b)) inter-die skewed process point for different levels of available redundancy. The memory is assumed to comprise 1024 rows with each row arranged as 4 columns of 32 bytes (see Figure 5.9). We have used equations (5.19) and (5.21) to compute the yield of the memory at various redundancy ratios, i.e., $r = R/M$.

Average leakage power consumption of the memory as a function of the standby voltage is also shown in Figure 5.6. As can be seen, subject to an identical target yield, the standby voltage can be reduced more aggressively for the dies with the nominal Vt compared to those with skewed process parameters. For example, without redundancy, the standby voltage of a nominal-Vt die (Figure 5.6(a)) can be reduced down to $0.58V$, and its leakage is cut by 62% while keeping the array failure probability below 1%. Whereas, for the die with skewed process parameters (Figure 5.6(b)), the standby voltage can only be reduced down to $0.89V$ without redundancy, and leakage is reduced by only 16%.

Devising redundancy resources allows the standby voltage of the memory to be



(a) (b)

Figure 5.6. Array failure probability of a 1Mb memory versus standby voltage with different levels of available redundancy ($r = R/M$) at (a) nominal-Vt corner, and (b) $\Delta Vt_{inter} = -100mV$.

reduced further compared to a memory without redundancy, while still meeting the same target yield. Adding $r = 1\%$ redundancy, for example, allows the standby voltage to be reduced to $0.44V$ and $0.69V$ for the nominal-Vt and skewed dies, respectively. Thereby, their leakage power can be cut down more effectively, i.e., 70% and 36% for nominal-Vt and FS dies, respectively. However, as can be seen from Figure 5.6, increasing redundancy resources beyond $r = 1\%$ allows only a marginal reduction in the standby voltage of the memory and thus loses its efficiency. This is because of a sharp increase in the probability of DRFs with the reduction of supply voltage (see Figure 5.4). Hence, we consider only a redundancy ratio of $r = 1\%$ in the rest of our analysis.

5.5.3 Impact of Inter-die Variations on Data-Retention Yield

Array failure probability at a given standby voltage varies for dies with different skewed process parameters. For example, the failure probability of a 1Mb memory array at various inter-die skewed threshold voltages is shown in Figure 5.7 for two

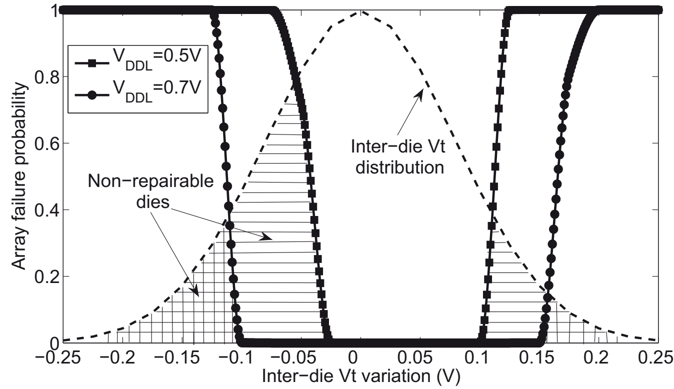


Figure 5.7. Dies at some skewed process points become non-repairable by the available redundancy ($r = 1\%$), when an identical standby voltage is applied to all dies.

different standby voltages. The organization of the memory is assumed to be as shown in Figure 5.9, and the level of the available redundancy is set to $r = 1\%$. Inter-die variation of Vt, modeled as a normal distribution $N(0, \sigma_{Vt_{inter}})$, is superimposed to illustrate the relative population of arrays at various inter-die skewed process points. If $V_{DDL} = 0.7V$ is applied to this memory, for example, then some of the dies, those in the region filled with vertical lines in Figure 5.7, will not be repairable due to a large number of DRFs. Decreasing the standby voltage to $V_{DDL} = 0.5V$ results in the failure of more dies as illustrated by the region filled with horizontal lines.

The overall array failure probability can be calculated as the weighted mean of the array failure probabilities at various skewed Vt points, i.e.,

$$P_{f,arr}(V_{DDL}) = \int_{-\infty}^{+\infty} P_{f,arr}(Vt_{inter}) N(Vt_{inter}) dVt_{inter} \tag{5.23}$$

where N is a discrete normal distribution function, i.e., $N \sim N(Vt_{nom}, \sigma_{Vt_{inter}})$.

5.5.4 Impact of Process Parameters Variations on Leakage Yield

Due to an exponential dependence of subthreshold leakage on the threshold voltage, the leakage of different cells in a memory array can be modeled as independent log-normal variables [68]. The total leakage of a memory array is the summation of the leakage of its constituent cells. Thus, using the central limit theorem, the distribution of the total memory leakage can be approximated as a Gaussian with mean (μ_{arr}) and standard deviation (σ_{arr}) given by [68]

$$\mu_{arr} = N\mu_{cell} \quad \text{and} \quad \sigma_{arr} = \sqrt{N}\sigma_{cell} \tag{5.24}$$

where, N is the total number of cells in the array, and μ_{cell} and σ_{cell} are the mean and standard deviation of the intra-die leakage distribution of individual cells.

Inter-die variations in Vt, on the other hand, result in a large spread in the mean leakage of the arrays [68]. This can cause some of the SRAM dies to have a leakage larger than the tolerable limit (I_{Lmax})). Such dies should be discarded as they violate the power budget, resulting in a yield loss due to high leakage. The probability that leakage of a die is less than the I_{Lmax} is given by

$$P(I_{L,arr} < I_{Lmax}) = \Phi\left(\frac{I_{Lmax} - \mu_{arr}}{\sigma_{arr}}\right) \tag{5.25}$$

Thus, the yield leakage can be defined as

$$Y_L = \int_{-\infty}^{+\infty} P\left(I_{L,arr}(Vt_{inter}) < I_{Lmax}\right) N(Vt_{inter}) dVt_{inter} \tag{5.26}$$

At each skewed inter-die process point, we performed MC simulations and swept the V_{DDL} from 1V to 0 by a $10mV$ decrement to obtain the intra-die distribution of cell leakages as a function of V_{DDL}. Then, the μ_{cell} and σ_{cell} are calculated at each process point and V_{DDL}. Using (5.24), we calculated the mean and standard deviation of array leakage distributions. Then, yield leakage is calculated using (5.26).

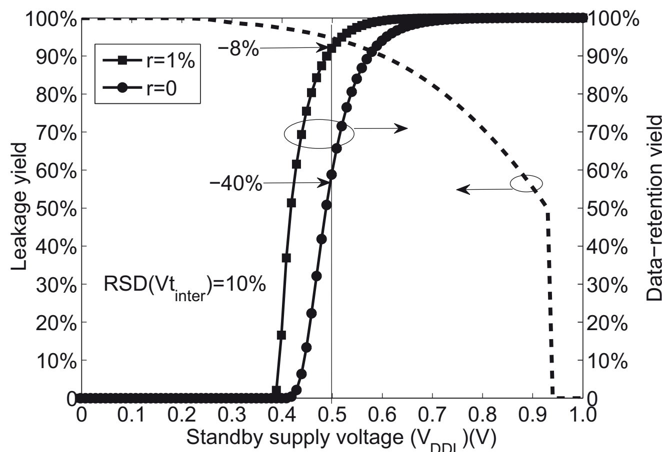


Figure 5.8. Data-retention and leakage yield versus standby voltage.

5.5.5 Tradeoff Between Data-Retention and Leakage Yield

Figure 5.8 shows the data-retention and leakage yield of a 1Mb memory versus standby voltage at various levels of available redundancy, assuming a 10% relative standard deviation (RSD) for inter- and intra-die Vt variations, i.e., $\sigma_{Vt_{inter}} = \sigma_{Vt_{intra}} = 0.1Vt_{nom}$ ($\sim$ 14% total variation). The maximum allowable leakage is assumed to be $150mW$. The tradeoff between leakage reduction and yield of SRAMs is clearly revealed from this figure. As can be seen, reducing the standby voltage increases the yield leakage. However, it also results in a sharp decrease of the data-retention yield. For example, if an identical standby voltage of $V_{DDL} = 0.5V$ is applied to all the memory dies to obtain a 95% leakage yield, yield losses due to the DRFs rises as high as 40% and 8% with $r = 0$ and $r = 1\%$ redundancy, respectively. On the other hand, if the standby voltage of dies is determined so that a 99% target data-retention yield is met, it can only be reduced down to $0.7V$, incurring $\sim$ 20% yield loss due to dies with excess leakage. The leakage yield losses can be reduced down to $\sim$ 10% by adding 1% redundancy and reducing the standby voltage to $0.58V$.

5.6 Yield Enhancement by Standby Supply Voltage Tuning

As shown in the previous section, applying an identical standby voltage to all memory dies regardless of their specific process parameters variation can drastically impact yield. In the following, we investigate the possibility of a post-silicon tuning technique to enable adjusting the standby voltage of each individual die to its corresponding V_{DDLmin}.

5.6.1 Post-Silicon Standby Voltage Tuning

To tune the standby supply voltage of each individual SRAM die to its minimum, the following features are required:

1. A procedure to identify V_{DDLmin} of each individual die after fabrication.
2. A circuit to generate and apply that V_{DDLmin} to each memory die (or embedded memory module) during the standby mode.

In the following, we investigate these two requirements.

5.6.1.1 Identifying V_{DDLmin} of a Memory Die

V_{DDLmin} of a memory die is the minimum standby voltage at which the die still functions correctly. It is determined by the specific inter- and intra-die process parameters variations which have happened in that particular die. Hence, it can only be identified after a memory die is fabricated. We propose a test procedure to search for the minimum standby voltage at which a memory die still functions free of fault. We choose a simple search algorithm in which the standby supply voltage of the memory-

under-test is gradually reduced from nominal V_{DD} by a small decrement (Δ), and then a test procedure is performed at each standby voltage to check for errors. The only faults that may be sensitized by switching a memory to the standby mode are data-retention faults. Thus, the proposed test procedure checks only for the '0' and '1' DRFs.

For a memory with no redundancy, if faulty cells are detected at a certain standby voltage, the applied standby voltage in the previous iteration is determined as its V_{DDLmin}. However, a memory with redundancy can be repaired as long as the number of DRFs is small enough. Assuming a row redundancy, those rows that contain faulty cell(s) can be replaced by the redundant rows. As the standby voltage is reduced, the number of faulty rows will increase. Thus, V_{DDLmin} of a memory with redundancy is determined as the lowest V_{DD} at which the number of failing rows does not exceed the number of redundant rows.

A detailed calibration procedure is proposed in Algorithm 1. Starting from $V_{DDL} = V_{DD}$, memory is first checked for '0' data-retention failures, by writing all '0' to the memory and then putting it in the standby mode for an adequate time p. The maximum pause required for triggering data-retention failures was found to be $\sim 2ms$. Then, memory is awakened and all '0' is read from all cells. The above test is repeated to check for '1' data-retention failures as well, by writing all '1' to the memory. This procedure is continued with progressively lower V_{DDL} until the number of failing rows becomes larger than the number of redundant rows at a given step. Then, the corresponding standby voltage of the previous iteration is marked as the V_{DDLmin} of this memory.

The test time required to detect the V_{DDLmin} of dies by the proposed algorithm depends on the supply voltage step (Δ) and the number of memory locations (n). At each step, n read, n write, and one pause operation are performed first with '0' and

Algorithm 1 Post-silicon standby supply voltage tuning

Require: No. of redundant rows (R), Standby voltage resolution (Δ)

1: $V_{DDL} \Leftarrow V_{DD}$;
2: **while** $V_{DDL} \geq 0$ **do**
3: write '0' to all cells;
4: switch to standby mode; pause for p seconds;
5: switch to active mode; read all '0' from all cells;
6: **if** No. of failing rows $> R$ **then**
7: break;
8: **end if**
9: write '1' to all cells;
10: switch to standby mode; pause for p seconds;
11: switch to active mode; read '1' from all cells;
12: **if** No. of failing rows $> R$ **then**
13: break;
14: **end if**
15: $V_{DDL} \Leftarrow V_{DDL} - \Delta$;
16: **end while**

Ensure: $V_{DDLmin} \Leftarrow V_{DDL} + \Delta$;

then with '1' data backgrounds. Thus, the upper bound of the test time is given by $2(2nT + p)\frac{V_{DD}}{\Delta}$, where T is the access time and p is the pause time. The test time grows with the increase in the tuning resolution, i.e., small Δ. Thus, a more complex search algorithm might be used instead of the proposed linear search in Algorithm 1.

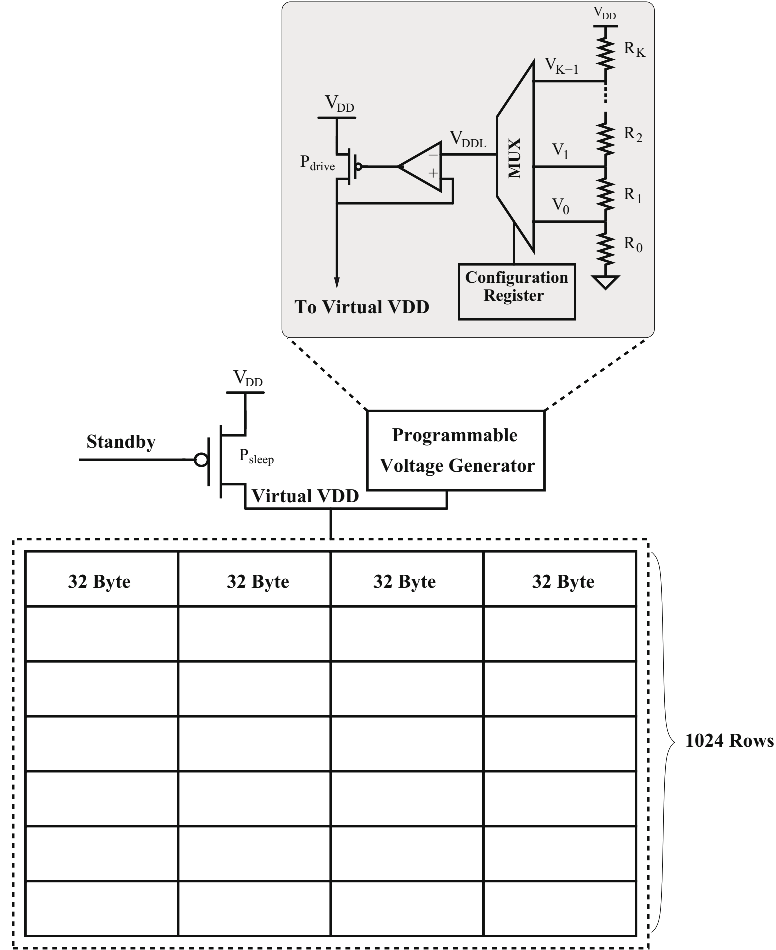


Figure 5.9. Standby supply voltage tuning scheme.

5.6.1.2 Generating V_{DDLmin} and Applying it to a Memory Array

The identified V_{DDLmin} of a memory can be generated on-chip using a programmable voltage generator as in [77]. Figure 5.9 shows a possible implementation of such a programmable voltage generator. A resistive divider is used to generate K reference voltages on-chip. Then, an analog multiplexer (MUX) is used to select the voltage that is going to be applied as the standby voltage to the memory array during the standby modes. The select inputs of the MUX are driven by a configuration register which is permanently programmed at test time, e.g., through fuses, so that

the voltage closest to the identified V_{DDLmin} is selected by the MUX and applied to the memory during the life-time of the chip.

A voltage follower circuit, see Figure 5.9, is used to actively tune the supply voltage of the block to the corresponding V_{DDLmin} during standby modes. When the *Standby* signal is set high, by a power management unit for example, in order to put the memory in standby mode, the transistor P_{sleep} is turned off, thus isolating the power supply from the memory. At this point, the virtual V_{DD} node, which is generally a large decoupling capacitance, starts to discharge due to the leakage currents of the memory array. However, it is never allowed to fall below V_{DDLmin} of the array by the voltage follower circuit. The voltage-follower exploits a simple feedback mechanism that monitors the voltage of the virtual V_{DD} node and pumps charge into it through the PMOS transistor (P_{drive}), if it falls below V_{DDL}. When the array is awakened by deactivating the *Standby* signal, the large sleep transistor (P_{Sleep}) is turned on and the virtual V_{DD} node is tightly connected to the supply node, rapidly charging it to the nominal V_{DD}. As V_{DD} is larger than the reference voltages, the voltage follower circuit does not interfere with the normal operation of the memory during the active mode, i.e., when $Standby =$'0'.

5.6.2 Overhead of the Tuning Technique

5.6.2.1 Area Overhead

The major area overhead of the proposed tuning technique is due to the large PMOS sleep transistor (P_{sleep}) and the PMOS drive transistor (P_{drive}) (see Figure 5.9). The P_{sleep} transistor needs to be sized large enough to avoid write time penalties and also to reduce the wake-up latency of the memory. The P_{drive} transistor, on the other hand, needs to be large enough to be able to provide data-retention currents of the

memory block during standby mode. Our experiments with actual layout of a large SRAM block show that the area overhead of the proposed technique can be kept below 2%.

5.6.2.2 Power Overhead

In order for the proposed technique to be beneficial, the static power consumption of the tuning circuitry must be negligible. The leakage power of the resistive voltage divider can be made very small by using large resistances. The MUX and the configuration register can be designed using high threshold transistors, making their leakage power negligible. Thus, the major source of the static power overhead in Figure 5.9 is the bias current of the op-amp. The power overhead of the tuning circuit can be significant for small memory arrays. However, as the size of memory grows, the relative overhead of the tuning technique decreases accordingly. Therefore, for large memories, we expect that the overhead of the technique would be negligible.

5.7 Simulation Results for Yield Enhancements and Discussions

In this section, we present analytical and simulation results on yield enhancements that can be obtained by the proposed standby voltage tuning technique. We first derive the distribution of V_{DDLmin} of memory arrays that might be found by the proposed algorithm in a typical inter-die variations setup.

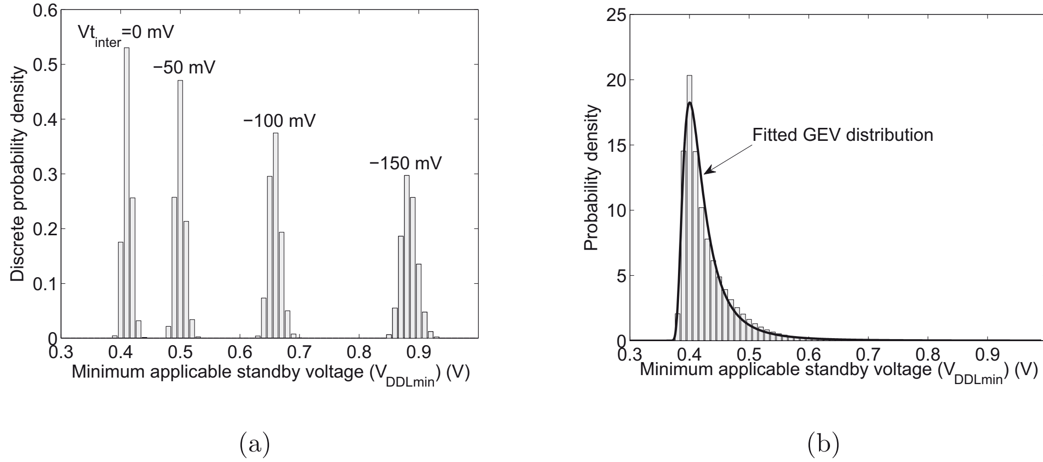


Figure 5.10. (a) Distribution of V_{DDLmin} at various inter-die corners, and (b) Distribution of V_{DDLmin} for memory dies in a process with $\sigma_{Vt_{inter}} = 10\%$.

5.7.1 V_{DDLmin} Distribution

The proposed tuning technique results in a distribution of the standby voltages of memory dies or modules. Using (5.12), the CDF of V_{DDLmin} can be calculated based on the array failure probabilities, i.e.,

$$Pr(V_{DDLmin} \leq V_{DDL}) = 1 - P_{f,arr}(V_{DDL}). \tag{5.27}$$

Figure 5.10(a) shows the distribution of V_{DDLmin} at various inter-die threshold voltage setups, assuming $\Delta = 10mV$. As can be seen, variation in the V_{DDLmin} of dies which have the same skewed process parameters setup is very small. However, V_{DDLmin} varies significantly among dies at different skewed process points. Given the overall array failure probability, the distribution of V_{DDLmin} due to the inter-die variations, can also be calculated using (5.27). We use (5.23) to estimate the overall array failure probability as a function of V_{DDL}. Figure 5.10(b) shows the distribution of V_{DDLmin} of dies identified by Algorithm 1 with a voltage step equal to $\Delta = 10mV$. The relative standard deviation (RSD) of the threshold voltage is set to 10% (i.e., $\sigma_{Vt_{inter}} = 0.1Vt_{nom}$).

As explained in Subsection 5.2.4, the inter-die distribution of V_{DDLmin} should

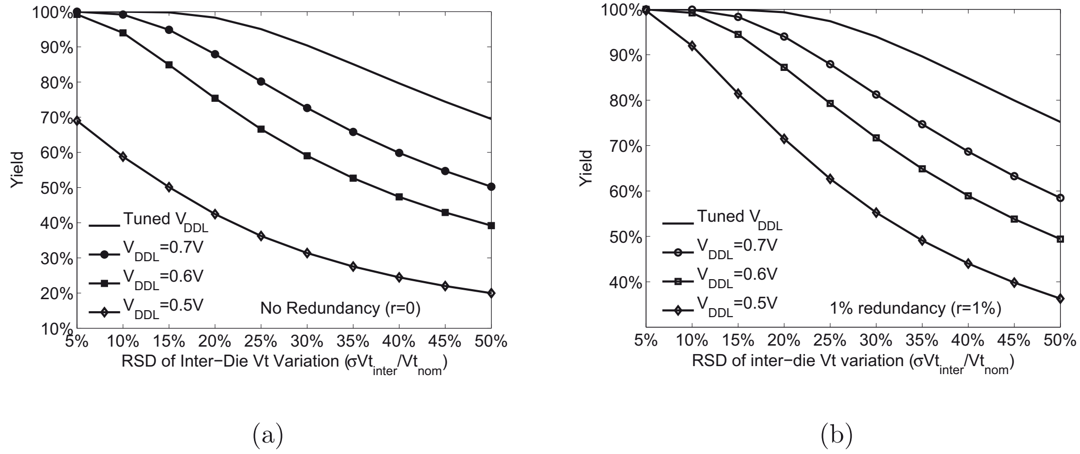


Figure 5.11. (a) Yield of a 1Mb memory versus relative standard deviation (RSD) of inter-die Vt variation at fixed and tuned standby voltages with no redundancy, and (b) with 1% redundancy ratio.

converge to a GEV distribution according to the EVT. To examine this, we fitted a GEV distribution to the simulation data using maximum likelihood estimation (MLE) methods. The fitted GEV distribution is shown in Figure 5.10(b) with a solid line. The estimated shape parameter of the GEV distribution, i.e., $\xi \approx 0.35$, indicates that V_{DDLmin} converges in distribution to a Frechet distribution. This means that, the inter-die distribution of V_{DDLmin} is bounded on the lower side and has a heavy right tail. The heavy tail of the V_{DDLmin} distribution at large standby voltages implies that yield predictions based on an assumption of a thin tail V_{DDLmin} distribution, e.g., a normal distribution, can result in too optimistic results.

5.7.2 Yield Enhancements by Standby Voltage Tuning

Tuning the standby voltage of memory dies (or modules) to their corresponding V_{DDLmin} prevents failures due to DRFs during the standby mode. We calculated the yield of a 1Mb memory array, with an organization as shown in Figure 5.9, using (5.23) and (5.11). The inter-die distribution of Vt is assumed to be a normal distribution.

In order to investigate the yield losses at various degrees of variations, we sweep the relative standard deviation (RSD) of the inter-die Vt distribution from 5% to 50%.

Figure 5.11 shows the yield of the 1Mb memory array versus relative standard deviation (RSD) of the inter-die threshold voltage distribution, at two different levels of redundancy. The yield is calculated at three different fixed standby voltages as well as the case when V_{DDL} of dies is tuned. For the memory with no redundant resources (Figure 5.11(a)), applying an identical standby voltage to all dies results in drastic yield losses even at low degrees of variations. For example, at $RSD(Vt_{inter}) = 10\%$, reducing standby voltage to $V_{DDL} = 0.5V$ results in about 40% yield loss. Increasing the standby voltage to $V_{DDL} = 0.7V$ eliminates the yield losses, but at the expense of larger leakage power dissipation. However, at high variations, $RSD(Vt_{inter}) = 30\%$ for example, increasing the standby voltage from $0.5V$ to $0.7V$ can only enhance yield to $\sim 75\%$. By tuning the standby voltage of dies, however, a large portion of failing dies are salvaged and thus the yield increases to $\sim 90\%$. Note that the yield losses in the standby voltage tuning scheme are due to the dies that fail even when their standby voltage is tuned to the nominal V_{DD}, and thus the proposed scheme is not capable of salvaging them. Therefore, these parametric yield losses are not due to the introduction of the standby mode to the memories, as they will fail even in the active mode due to their highly skewed process parameters.

Adding redundancy enhances yield at all variation scenarios as shown in Figure 5.11(b). For example, when an identical standby voltage of $V_{DDL} = 0.5V$ is applied to all dies, yield is enhanced from 70% to 100% at $RSD(Vt_{inter}) = 5\%$ by adding only $r = 1\%$ redundancy. This is due to the low probability of DRFs, which allows the arrays to be repaired. However, if Vt variations deteriorate, yield cannot be enhanced remarkably by adding only redundancy. For example, at $RSD(Vt_{inter}) = 20\%$ and $V_{DDL} = 0.5V$, adding redundancy can only improve yield by 30%, i.e., from 42% to 72%. Tuning the standby voltage of dies, however, allows a much better yield

to be obtained even at very extreme variations of threshold voltage. For example, at $RSD(Vt_{inter}) = 20\%$, yield is improved to $\sim 99\%$ by the combined effect of redundancy and standby voltage tuning. Therefore, our proposed post-silicon standby voltage tuning scheme can be more beneficial at nano-scale technologies, where the variation of process parameters is expected to deteriorate.

5.7.3 Yield Losses Due to Dies With Excess Leakage in Case of Voltage Tuning

The V_{DDLmin} of some dies in the proposed tuning techniques will be higher than the threshold that is required to meet a predefined leakage yield. Hence, it is expected that the leakage yield reduces in case of tuning. However, the increases in the standby voltage of these dies is not very large. For example, for dies at $\Delta Vt_{inter} = -100mV$ and 1% redundancy, raising the standby voltage by only 0.1V (from 0.6V to 0.7V) removes all the yield losses due to the DRFs (see Figure 5.6(b)). Moreover, the leakage increases very slowly with the raising of V_{DDL} in this region as can be seen in Figure 5.6(b). Therefore, the leakage yield loss incurred by tuning the standby voltage of dies will be negligible.

5.7.4 Uncorrelated Inter-Die Shift for NMOS and PMOS

Our assumption of a "-1" correlation coefficient between the threshold voltage shift of NMOS and PMOS transistors results in pessimistic estimates for the cell failure probability. A shift in Vt of NMOS and PMOS transistors in the same direction causes less cell imbalance than the anti-correlated shift. Therefore, the cell failure probability is expected to be the highest along the anti-correlated axes in the Vt variation plane (see Figure 5.3). The amount of variance in our analysis is a parameter swept from

$5\% - 50\%$. This parameter is the variance of the projected one-dimensional Gaussian distribution, and is smaller than the true amount of inter-die variance. Therefore, the reported failure probability estimates in this work correspond to higher true inter-die variations. The amount of incurred pessimism remains to be explored by performing a two-dimensional analysis of the problem based on the exact joint distribution of the inter-die Vt variations.

5.8 Summary

A post-silicon standby supply voltage tuning scheme for SRAMs was presented to decrease yield losses due to parametric data-retention failures during the standby mode, while reducing the leakage currents effectively. It was shown that applying an identical standby voltage to all dies, regardless of their specific process parameters variations, can result in the failure of some dies, due to data-retention failures, and thus it entails significant yield losses. To avoid yield losses, we proposed to tune the standby voltage of each individual die to its corresponding minimum level. A test algorithm was presented to identify the minimum applicable standby voltage to each individual memory die after manufacturing. The effect of adding redundant resources on the minimum applicable standby voltage to a memory die was also investigated. Simulation results in a 45-nm predictive technology showed that yield can be enhanced significantly by the combined effect of repairing and standby voltage tuning, even when heavy process variations are present.

Chapter 6

Conclusions and Future Work

A low power and robust design of SRAMs is crucial for the overall success of modern microprocessors and SoCs, as they occupy the majority of the chip area in a wide range of applications. With technology scaling down to nano-meter feature sizes, satisfying the multi-dimensional requirements of low power and high yield for SRAMs is becoming increasingly difficult, due to the generally contradictory nature of these design requirements. In particular, reducing the power consumption of SRAMs while maintaining the full functionality across the whole array is becoming increasingly challenging. That is because the power reduction techniques, such as voltage scaling, also degrade the cells robustness, and thereby result in the failure of an increasingly larger number of cells that are already weakened by excessive process parameters variations and/or manufacturing imperfections in nanometer technologies. In this research, we have performed a thorough analysis of the involved yield-power tradeoffs in SRAMs, and proposed solutions to address the design paradox of their joint low-power dissipation and high yield. The major contributions and possible future work of this research are summarized in the following sections.

6.1 Contributions and Main Results

We performed fault injection and simulation to investigate the fault behavior of open defects in SRAM core cells when they are switched to a drowsy operating mode. We showed that, in addition to the data-retention faults, open defects in SRAM cells can also result in faulty behaviors when a cell is accessed immediately after wake-up. We described these new read-after-drowsy (RAD) fault behaviors and derived their corresponding fault primitives (FPs). Then, we used the derived FPs to design a new March test by inserting drowsy operations to a traditional test algorithm. The proposed March test, called March RAD, is capable of detecting all drowsy faults as well as the simple traditional faults. Finally, it was shown that as the supply voltage is reduced to further cut down leakage, defects with smaller parasitic resistances start to be sensitized and cause failure. Thereby, the tradeoff between yield and leakage power of SRAMs was pointed out. The results from this part of our research were reported in [78].

Process parameters variations can also degrade the data-retention capability of SRAMs. In particular, it was shown that extreme process parameters variations can result in weak SRAM cells with marginal data-retention capability, so that even a moderate scaling of the supply voltage can result in their failure. Such extremal events were found to be very rare, however, their probability is magnified by the huge number of replicated bitcell on modern embedded memories. Hence, it is critical to also account for such extremal events while attempting to scale the supply voltage of SRAMs. To estimate the statistics of such rare failures in a reasonable computational time, we employed concepts from extreme value theory (EVT). In particular, a limited number of MC simulations were first performed to obtain a sufficient number of data points in the tail region of the cells minimum standby voltage distribution. Then, we employed the peak over threshold method to fit a generalized Pareto distribution to

the data points that exceeded a certain threshold. This enabled us to make predictions far out in the tail of the cell failure probability distribution, without having actual simulation data in those regions. Mathematical relations were then developed to compute the yield of a complete memory array based on the failure probability of a single cell. The results showed that even moderate voltage scalings can result in considerable yield losses in large SRAMs, due to the failure of the highly skewed bitcells. The results of this analysis are reported in [79].

Yield losses due to the DRFs can especially limit the leakage reduction of SRAMs in new technologies, as the process parameters variations are expected to deteriorate with technology scaling. Thus, we investigated the application of fault-tolerance techniques for a more efficient leakage reduction of SRAMs. These techniques allow for a more aggressive voltage scaling by providing tolerance to the failures that might occur during the sleep mode. The results showed that in a 45-nm technology, assuming 10% variation in transistors threshold voltage, repairing a 64KB memory using only 8 redundant rows or incorporating single error correcting Hamming codes allows for $\sim 90\%$ leakage reduction while incurring only $\sim 1\%$ yield loss, at the expense of $\sim 4\%$ area increase. The combination of redundancy and ECC, however, allowed to reach the ultimate bounds of the leakage reduction, i.e., $\sim 95\%$. Thus, the latter approach can be attractive in ultra-low leakage applications, especially when variations are large and the activity factor of memory is small, so that the overhead associated with the dynamic power of ECC encoding/decoding becomes negligible. These findings were reported in [79].

The fault-tolerance techniques can counter the failures within an array as long as their number is limited. However, it was shown that due to the inter-die variations, the probability of cell failures at a given supply voltage can vary significantly from die to die. Applying an identical standby voltage to all dies, regardless of their specific process parameters variations, was thus shown to render some dies unsalvageable by

the employed fault-tolerance techniques. To compensate for the inter-die variations, a post-silicon standby supply voltage tuning scheme for SRAMs was proposed that decreases yield losses due to the failure of dies with highly skewed process parameters. In this technique, we proposed to tune the standby voltage of each individual die to its corresponding minimum level after manufacturing. A test algorithm was presented to identify the minimum applicable standby voltage to each individual memory die. Simulation results in a 45-nm predictive technology showed that tuning standby voltage of SRAMs can enhance data-retention yield by an additional $10\% - 50\%$, depending on the severity of the variations. The results were reported in [80].

6.2 Future Work

Some possible future work on this research can be as follows.

6.2.1 More Efficient Tests for Detection of Drowsy Faults

The proposed March RAD algorithm contains two drowsy operations which require switching the whole memory array to the drowsy mode and keeping it in that mode for a certain period in order to detect drowsy faults. This procedure can be very time-consuming due to the switchings between active and drowsy modes and a long pause in the drowsy mode. In addition, the potential dynamic faults that require performing multiple operations in sequence to be sensitized remain undetected to March RAD. Such test escapes can degrade the defects-per-million (DPM) figure of the memory.

An analysis of the simulation results for drowsy faults in Section 3.4 reveals that the fault behavior of the PODs changes in the same fashion with the increase in defect resistance. That is, in the active mode, data retention faults (DRFs) are the

first faults to emerge as the resistance of a POD increases beyond a critical level. If the resistance of a POD falls below this critical level, it will result only in a weak fault in the active mode. However, in the drowsy mode, this same POD exhibits a static/dynamic drowsy fault. This implies that, the detection of drowsy faults might be translated to the detection of weak faults in the active mode. Several techniques have been proposed in the literature for the fast detection of weak PMOS devices caused by PODs, by means of stressing the SRAM cells [52, 81, 35, 82]. The programmability of the stress level in these techniques allows one to filter out only the cells weaker than a certain threshold. As a result, these techniques can be good candidates for a fast detection of the drowsy faults.

6.2.2 A Built-In Technique for Self-Tuning of Standby Supply Voltage Against Run-Time Variations

In the proposed post-silicon tuning scheme, we proposed to identify the minimum standby supply voltage of a memory die (V_{DDLmin}) once after manufacturing, and program it permanently on-chip, e.g., through fuses. However, the minimum standby supply voltage of a memory array can vary on the field. The short-term variations in V_{DDLmin} can be due to environmental variations, e.g., temperature variations. The long term skews in V_{DDLmin} can also happen due to the changing device parameters by aging, e.g., the threshold voltage shift due to negative bias temperature instability (NBTI) [83]. A built-in self-tuning technique can be used to trace for these variations and adapt the applied supply voltage to the memory during the run time.

6.2.3 Compensating for Systematic Intra-Die Variations

Data-retention voltage (DRV) of SRAM cells vary within a memory die due to both local random variations, e.g., due to random dopant fluctuations (RDF) [40, 34], and systematic intra-die variations, e.g., due to photo-lithographic and etching variations [37, 60]. Systematic variations exhibit a strong spatial correlation [37]. Thus they result in similar variations in the characteristics of neighboring cells, including V_{DDLmin}. Therefore, larger leakage reductions can be obtained by partitioning the memory into sufficiently small groups of cells and tuning the supply voltage of each individual group to its corresponding minimum. Our proposed standby supply voltage tuning technique can be extended to a sub-array level by associating a distinct programmable reference voltage generator (see Figure 5.9) to each sub-array. The configuration data for each sub-array can be identified in a similar way by Algorithm 1. The extra leakage reductions of this within-die voltage tuning technique increases as the size of the sub-array reduces. However, the overhead of the technique can be intractable at small granularity. Thus, the sub-array size needs to be chosen carefully. A good candidate is to tune the supply voltage of multiple embedded memory modules of sufficient size that are found in modern multi-processor system-on-chips (MPSoCs).

References

[1] J. Rabaey, *Low Power Design Essentials.* Springer, 2009.

[2] K. Zhang, *Embedded Memories for Nano-Scale VLSIs.* Springer, 2009.

[3] A. Pavlov and M. Sachdev, *CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies.* New York, USA: Springer, 2008.

[4] S. Rusu, S. Tam, H. Muljono, J. Stinson, D. Ayers, J. Chang, R. Varada, M. Ratta, S. Kottapalli, and S. Vora, "A 45 nm 8-core enterprise Xeon® processor," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 1, pp. 7 –14, jan. 2010.

[5] *International Technology Roadmap for Semiconductors*, Semiconductor Industry Association, 2009. [Online]. Available: http://www.itrs.net

[6] R. Krishnarnurthy, A. Alvandpour, V. De, and S. Borkar, "High-performance and low-power challenges for sub-70 nm microprocessor circuits," in *Custom Integrated Circuits Conference, 2002. Proceedings of the IEEE 2002*, 2002, pp. 125 – 128.

[7] C. Kenyon, A. Kornfeld, K. Kuhn, M. Liu, A. Maheshwari, W. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki, "Managing process variation in Intel's 45nm CMOS technology," *Intel Technology Journal*, vol. 12, no. 2, pp. 93–109, June 2008. [Online]. Available: http://www.intel.com/technology/itj/2008/v12i2/3-managing/1-abstract.htm

[8] H. Yamauchi, "A discussion on SRAM circuit design trend in deeper nanometer-scale technologies," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 5, pp. 763 –774, may 2010.

[9] Y. Ye, F. Liu, M. Chen, S. Nassif, and Y. Cao, "Statistical modeling and simulation of threshold variation under random dopant fluctuations and line-edge roughness," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –10, 2010.

[10] S. Nassif, "Modeling and analysis of manufacturing variations," in *Custom Integrated Circuits, 2001, IEEE Conference on.*, 2001, pp. 223 –228.

[11] R. Montanes, J. de Gyvez, and P. Volf, "Resistance characterization for weak open defects," *Design & Test of Computers, IEEE*, vol. 19, no. 5, pp. 18–26, Sep-Oct 2002.

[12] J. M. Rabaey, A. Chandrakasan, , and B. Nikolic, *Digital Integrated Circuits- A Design Perspective, Second Edition.* Prentice-Hall, 2003.

[13] L. Chang, R. Montoye, Y. Nakamura, K. Batson, R. Eickemeyer, R. Dennard, W. Haensch, and D. Jamsek, "An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 4, pp. 956 –963, april 2008.

[14] N. Verma and A. Chandrakasan, "A 65nm 8T sub-Vt SRAM employing sense-amplifier redundancy," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, 11-15 2007, pp. 328 –606.

[15] I. J. Chang, J.-J. Kim, S. Park, and K. Roy, "A 32 kb 10t sub-threshold SRAM array with bit-interleaving and differential read scheme in 90 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 2, pp. 650 –658, feb. 2009.

[16] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, Feb 2003.

[17] Predictive Technology Model. [Online]. Available: http://ptm.asu.edu/

[18] F. Hamzaoglu, K. Zhang, Y. Wang, H. Ahn, U. Bhattacharya, Z. Chen, Y.-G. Ng, A. Pavlov, K. Smits, and M. Bohr, "A 3.8 GHz 153 Mb SRAM design with dynamic stability enhancement and leakage reduction in 45 nm high-k metal gate CMOS technology," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 148–154, Jan. 2009.

[19] N. S. Kim, K. Flautner, D. Blaauw, and T. Mudge, "Circuit and microarchitectural techniques for reducing cache leakage power," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 12, no. 2, pp. 167–184, Feb. 2004.

[20] H. Hanson, M. Hrishikesh, V. Agarwal, S. Keckler, and D. Burger, "Static energy reduction techniques for microprocessor caches," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 303–313, June 2003.

[21] A. Agarwal, H. Li, and K. Roy, "A single-Vt low-leakage gated-ground cache for deep submicron," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 2, pp. 319–328, Feb 2003.

[22] J. Chang, M. Huang, J. Shoemaker, J. Benoit, S.-L. Chen, W. Chen, S. Chiu, R. Ganesan, G. Leong, V. Lukka, S. Rusu, and D. Srivastava, "The 65-nm 16-MB shared on-die L3 cache for the dual-core Intel Xeon processor 7100 series," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 4, pp. 846–852, April 2007.

[23] H. Mizuno and T. Nagano, "Driving source-line cell architecture for sub-1-V high-speed low-power applications," *Solid-State Circuits, IEEE Journal of*, vol. 31, no. 4, pp. 552 –557, apr 1996.

[24] C. Kim and K. Roy, "Dynamic Vt SRAM: a leakage tolerant cache memory for low voltage microprocessors," in *Low Power Electronics and Design, 2002. ISLPED '02. Proceedings of the 2002 International Symposium on*, 2002, pp. 251 – 254.

[25] Y. Wang, H. J. Ahn, U. Bhattacharya, Z. Chen, T. Coan, F. Hamzaoglu, W. Hafez, C.-H. Jan, P. Kolar, S. Kulkarni, J.-F. Lin, Y.-G. Ng, I. Post, L. Wei, Y. Zhang, K. Zhang, and M. Bohr, "A 1.1 Ghz 12 μ A/Mb-leakage SRAM design in 65 nm ultra-low-power CMOS technology with integrated leakage reduction for mobile applications," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 172–179, Jan. 2008.

[26] T.-H. Kim, J. Liu, and C. Kim, "A voltage scalable 0.26 V, 64 kb 8T SRAM with V_{min} lowering techniques and deep sleep mode," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 6, pp. 1785–1795, June 2009.

[27] J. Tschanz, S. Narendra, Y. Ye, B. Bloechel, S. Borkar, and V. De, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 11, pp. 1838 – 1845, nov. 2003.

[28] A. Nourivand, C. Wang, and M. Omair Ahmad, "An adaptive sleep transistor biasing scheme for low leakage SRAM," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, 27-30 2007, pp. 2790 –2793.

[29] Y. Meng, T. Sherwood, and R. Kastner, "Exploring the limits of leakage power reduction in caches," *ACM Trans. Archit. Code Optim.*, vol. 2, no. 3, pp. 221–246, 2005.

[30] M. Sharifkhani and M. Sachdev, "Segmented virtual ground architecture for low-power embedded SRAM," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 15, no. 2, pp. 196 –205, feb. 2007.

[31] K.-S. Min, K. Kanda, and T. Sakurai, "Row-by-row dynamic source-line voltage control (RRDSV) scheme for two orders of magnitude leakage current reduction of sub-1-V-VDD SRAM's," in *Low Power Electronics and Design, 2003. ISLPED '03. Proceedings of the 2003 International Symposium on*, Aug. 2003, pp. 66–71.

[32] E. Seevinck, F. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *Solid-State Circuits, IEEE Journal of*, vol. 22, no. 5, pp. 748–754, Oct 1987.

[33] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T. Vijaykumar, "Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories," *Low*

Power Electronics and Design, 2000. ISLPED '00. Proceedings of the 2000 International Symposium on, pp. 90–95, 2000.

[34] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 24, no. 12, pp. 1859–1880, Dec. 2005.

[35] A. Pavlov, M. Sachdev, and J. De Gyvez, "Weak cell detection in deep-submicron SRAMs: A programmable detection technique," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 10, pp. 2334–2343, Oct. 2006.

[36] J. Segura, A. Keshavarzi, J. Soden, and C. Hawkins, "Parametric failures in CMOS ICs - a defect-based analysis," in *Test Conference, 2002. Proceedings. International*, 2002, pp. 90–99.

[37] K. Agarwal and S. Nassif, "Characterizing process variation in nanometer CMOS," in *DAC '07: Proceedings of the 44th annual Design Automation Conference*. New York, NY, USA: ACM, 2007, pp. 396–399.

[38] K. Bowman, S. Duvall, and J. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 2, pp. 183 –190, feb 2002.

[39] P. Stolk, F. Widdershoven, and D. Klaassen, "Modeling statistical dopant fluctuations in MOS transistors," *Electron Devices, IEEE Transactions on*, vol. 45, no. 9, pp. 1960 –1971, sep. 1998.

[40] A. Bhavnagarwala, X. Tang, and J. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *Solid-State Circuits, IEEE Journal of*, vol. 36, no. 4, pp. 658–665, Apr 2001.

[41] H. Qin, Y. Cao, D. Markovic, A. Vladimirescu, and J. Rabaey, "SRAM leakage suppression by minimizing standby supply voltage," *Quality Electronic Design, 2004. Proceedings. 5th International Symposium on*, pp. 55–60, 2004.

[42] J. Wang, A. Singhee, R. Rutenbar, and B. Calhoun, "Statistical modeling for the minimum standby supply voltage of a full SRAM array," in *Solid State Circuits Conference, 2007. ESSCIRC 2007. 33rd European*, Sept. 2007, pp. 400–403.

[43] A. Kumar, H. Qin, P. Ishwar, J. Rabaey, and K. Ramchandran, "Fundamental data retention limits in SRAM standby - experimental results," in *Quality Electronic Design, 2008. ISQED 2008. 9th International Symposium on*, March 2008, pp. 92–97.

[44] ——, "Fundamental bounds on power reduction during data-retention in standby SRAM," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, May 2007, pp. 1867–1870.

[45] H. Qin, A. Kumar, K. Ramchandran, J. Rabaey, and P. Ishwar, "Error-tolerant SRAM design for ultra-low power standby operation," in *ISQED '08: Proceedings of the 9th international symposium on Quality Electronic Design*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 30–34.

[46] S. Hamdioui and A. Van De Goor, "An experimental analysis of spot defects in SRAMs: realistic fault models and tests," *Test Symposium, 2000. (ATS 2000). Proceedings of the Ninth Asian*, pp. 131–138, 2000.

[47] A. Singhee and R. Rutenbar, "Statistical blockade: Very fast statistical simulation and modeling of rare circuit events and its application to memory design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 8, pp. 1176–1189, Aug. 2009.

[48] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events," in *Design Automation Conference, 2006 43rd ACM/IEEE*, 0-0 2006, pp. 69–72.

[49] L. Ding and P. Mazumder, "The impact of bit-line coupling and ground bounce on cmos sram performance," in *VLSI Design, 2003. Proceedings. 16th International Conference on*, jan. 2003, pp. 234 – 239.

[50] J. Yang, B. Wang, Y. Wu, and A. Ivanov, "Fast detection of data retention faults and other SRAM cell open defects," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 167–180, Jan. 2006.

[51] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel, S. Borri, and M. Hage-Hassan, "Resistive-open defects in embedded-SRAM core cells: analysis and march test solution," *Test Symposium, 2004. 13th Asian*, pp. 266–271, Nov. 2004.

[52] T. Mak, D. Bhattacharya, C. Prunty, B. Roeder, N. Ramadan, J. Ferguson, and J. Yu, "Cache RAM inductive fault analysis with fab defect modeling," *Test Conference, 1998. Proceedings., International*, pp. 862–871, Oct 1998.

[53] A. van de Goor and Z. Al-Ars, "Functional memory faults: a formal notation and a taxonomy," *VLSI Test Symposium, 2000. Proceedings. 18th IEEE*, pp. 281–289, 2000.

[54] W. Pei, W.-B. Jone, and Y. Hu, "Fault modeling and detection for drowsy SRAM caches," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 6, pp. 1084–1100, June 2007.

[55] P. Embrechts, T. Mikosch, and C. Klüppelberg, *Modelling Extremal Events: For Insurance and Finance*. London, UK: Springer-Verlag, 1997.

[56] H. Qin, Y. Cao, D. Markovic, A. Vladimirescu, and J. Rabaey, "SRAM leakage suppression by minimizing standby supply voltage," in *ISQED '04: Proceedings of the 5th International Symposium on Quality Electronic Design*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 55–60.

[57] S. Ghosh, S. Mukhopadhyay, K. Kim, and K. Roy, "Self-calibration technique for reduction of hold failures in low-power nano-scaled SRAM," in *DAC '06: Proceedings of the 43rd annual Design Automation Conference*. New York, NY, USA: ACM, 2006, pp. 971–976.

[58] A. Nourivand, A. Al-Khalili, and Y. Savaria, "Aggressive leakage reduction of SRAMs using error checking and correcting (ECC) techniques," in *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*, 10-13 2008, pp. 426 –429.

[59] A. Singhee, J. Wang, B. Calhoun, and R. Rutenbar, "Recursive statistical blockade: An enhanced technique for rare event simulation with application to SRAM circuit design," in *VLSI Design, 2008. VLSID 2008. 21st International Conference on*, jan. 2008, pp. 131 –136.

[60] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," in *Computer Aided Design, 2003. ICCAD-2003. International Conference on*, Nov. 2003, pp. 900–907.

[61] A. Agarwal, B. Paul, S. Mukhopadhyay, and K. Roy, "Process variation in embedded memories: failure analysis and variation aware architecture," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 9, pp. 1804 – 1814, sept. 2005.

[62] J. Hennessy and D. Patterson, *Computer Architecture - A Quantitative Approach, 3rd Edition*. San Mateo, CA: Morgan Kaufmann, 2003.

[63] C. Stapper and H.-S. Lee, "Synergistic fault-tolerance for memory chips," *Computers, IEEE Transactions on*, vol. 41, no. 9, pp. 1078–1087, Sep 1992.

[64] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: a state-of-the-art review," *IBM J. Res. Dev.*, vol. 28, no. 2, pp. 124–134, 1984.

[65] R. A. Fisher and L. H. C. Tippett, "Limiting forms of the frequency distribution of the largest or smallest member of a sample," *Proceedings of the Cambridge Philosophical Society*, vol. 44, no. 1, pp. 180–190, Apr. 1928.

[66] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 11, pp. 1396–1402, Nov 2002.

[67] J. Tschanz, S. Narendra, R. Nair, and V. De, "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 5, pp. 826–829, May 2003.

[68] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Reduction of parametric failures in sub-100-nm SRAM array using body bias," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 1, pp. 174–183, Jan. 2008.

[69] M. Khellah, D. Somasekhar, Y. Ye, N. S. Kim, J. Howard, G. Ruhl, M. Sunna, J. Tschanz, N. Borkar, F. Hamzaoglu, G. Pandya, A. Farhang, K. Zhang, and V. De, "A 256-Kb dual-VCC SRAM building block in 65-nm CMOS process with actively clamped sleep transistor," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 1, pp. 233 –242, jan. 2007.

[70] Y. Wang, U. Bhattacharya, F. Hamzaoglu, P. Kolar, Y.-G. Ng, L. Wei, Y. Zhang, K. Zhang, and M. Bohr, "A 4.0 GHz 291 Mb voltage-scalable SRAM design in a 32 nm high-k + metal-gate CMOS technology with integrated power management," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 1, pp. 103 –110, jan. 2010.

[71] S. Ghosh, S. Mukhopadhyay, K. Kim, and K. Roy, "Self-calibration technique for reduction of hold failures in low-power nano-scaled sram," in *Design Automation Conference, 2006 43rd ACM/IEEE*, 0-0 2006, pp. 971–976.

[72] N. Mojumder, S. Mukhopadhyay, J.-J. Kim, C.-T. Chuang, and K. Roy, "Self-repairing SRAM using on-chip detection and compensation," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 1, pp. 75 –84, jan. 2010.

[73] J. Wang and B. Calhoun, "Techniques to extend canary-based standby V_{DD} scaling for SRAMs to 45 nm and beyond," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 11, pp. 2514–2523, Nov. 2008.

[74] G. Chen, D. Sylvester, D. Blaauw, and T. Mudge, "Yield-driven near-threshold SRAM design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –1, 2009.

[75] B. Calhoun and A. Chandrakasan, "Static noise margin variation for subthreshold SRAM in 65-nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 7, pp. 1673–1679, July 2006.

[76] T. Hesterberg, "Weighted average importance sampling and defensive mixture distributions," *Technometrics*, vol. 37, no. 2, pp. 185–194, 1995. [Online]. Available: http://www.jstor.org/stable/1269620

[77] Y.-C. Lai, S.-Y. Huang, and H.-J. Hsu, “Resilient self-V_{DD}-tuning scheme with speed-margining for low-power SRAM,” *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 10, pp. 2817–2823, Oct. 2009.

[78] A. Nourivand, A. Al-Khalili, and Y. Savaria, “Analysis of resistive open defects in drowsy SRAM cells,” *submitted to the Journal of Electronic Testing: Theory and Applications (JETTA)*, 2010.

[79] ——, “Aggressive leakage reduction of SRAMs using fault-tolerance techniques: The yield-power tradeoff,” *submitted to the IEEE Transactions on Circuits and Systems I*, 2010.

[80] ——, “Post-silicon tuning of standby supply voltage in SRAMs to reduce yield losses due to parametric data-retention failures,” *accepted for publication in the IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2010.

[81] K. Zhang, U. Bhattacharya, Z. Chen, F. Hamzaoglu, D. Murray, N. Vallepalli, Y. Wang, B. Zheng, and M. Bohr, “SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction,” *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 4, pp. 895–901, April 2005.

[82] E. B. Selvin, A. R. Farhang, and D. A. Guddat, “Programmable weak write test mode,” US Patent 6 778 450, 2004.

[83] G. La Rosa, W. L. Ng, S. Rauch, R. Wong, and J. Sudijono, “Impact of NBTI induced statistical variation to SRAM cell stability,” in *Reliability Physics Symposium Proceedings, 2006. 44th Annual., IEEE International*, March 2006, pp. 274–282.