UC Santa Cruz
UC Santa Cruz Electronic Theses and Dissertations

Title
LOW-POWER METHODOLOGY FOR FAULT TOLERANT NANOSCALE MEMORY DESIGN

Permalink
https://escholarship.org/uc/item/1691s6q9

Author
Kim, Seokjoong

Publication Date
2012

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

LOW-POWER METHODOLOGY FOR FAULT TOLERANT NANOSCALE MEMORY DESIGN

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Seokjoong Kim

June 2012

The Dissertation of Seokjoong Kim
is approved:

Professor Matthew R. Guthaus, Chair

Professor Joel Ferguson

Professor Jose Renau

Dr. Krishnan Sundaresan

Dean Tyrus Miller
Vice Provost and Dean of Graduate Studies

Copyright © by

Seokjoong Kim

2012

Table of Contents

List of Figures vi

List of Tables viii

Abstract ix

Dedication xi

Acknowledgments xii

1 Introduction 1
- 1.1 Fault Tolerant Low-Power Memory . . . 2
- 1.2 Thesis Contributions and Outline . . . 3

2 Low-Power Memory Design 5
- 2.1 Static Random Access Memory (SRAM) . . . 5
  - 2.1.1 Memory Cell Structure (6T Cell) . . . 5
  - 2.1.2 Read/Write Operation . . . 6
  - 2.1.3 Static Noise Margin (SNM) . . . 9
- 2.2 Methodologies for Low-Power Memory . . . 10
  - 2.2.1 Current State of the Art: Toward Sub-Threshold SRAM . . . 10
  - 2.2.2 Voltage Scaling techniques . . . 11
  - 2.2.3 Transistor/Circuit level techniques . . . 14
  - 2.2.4 Cell Architecture Technique (8T Cell) . . . 17

3 Fault Tolerant Memory Design 19
- 3.1 Type of Errors (Hard Faults and Soft-Errors) . . . 19
  - 3.1.1 Hard Faults (Permanent Errors) . . . 19
  - 3.1.2 Soft-Errors (Single Event Upsets) . . . 20
- 3.2 Methodologies for Fault Tolerance . . . 23
  - 3.2.1 Redundant Row/Column for Repair Hard Faults . . . 23
  - 3.2.2 Various Techniques (ID, ECC and BICS) for Soft Errors . . . 25
  - 3.2.3 Dynamic Noise Margin (DNM) . . . 27

4 SNM-Aware Cell Optimization of 6T SRAMs 28
4.1 Motivation . . . 29
4.2 The Upperbound of PR . . . 30
4.3 SNM-Aware Cell Sizing . . . 30
4.4 Experimental Results . . . 32
4.4.1 Layout Area Overhead . . . 32
4.4.2 Performance . . . 33
4.4.3 Power Reduction . . . 33
4.4.4 Gains in Reliability . . . 34
4.5 Conclusions . . . 35

5 Leakage-Aware Redundancy for Reliable Sub-threshold Memories 37
5.1 Sub-threshold Reliability . . . 39
5.2 Yield Analysis Methodology . . . 42
5.2.1 Yield Simulation . . . 42
5.2.2 Optimal Fault Repair Analysis . . . 44
5.2.3 Leakage Calculation . . . 48
5.3 Leakage-Aware Redundancies . . . 50
5.3.1 Supply Voltage Lower Bound for Required Yield . . . 51
5.3.2 Leakage Optimization . . . 52
5.4 Supply Voltage for Active Mode Power . . . 54
5.4.1 Minimum Read/Write Voltage in Sub-threshold Regions . . . 55
5.4.2 Power Model using V_{stdby} and V_{act} . . . 58
5.5 Experimental Results . . . 60
5.5.1 Experimental Setup . . . 60
5.5.2 Yield Accuracy and CPU time . . . 61
5.5.3 Supply Voltage and Redundancy . . . 61
5.5.4 Leakage Reduction . . . 63
5.6 Conclusions . . . 66

6 Low-Power Multiple-Bit Upset Tolerant Memory Optimization 67
6.1 Transient Radiation Faults in Memory . . . 68
6.1.1 Soft Error Rate (SER) Model . . . 68
6.1.2 Multi-Bit Upset (MBU) Model ($\widehat{SER}$) . . . 69
6.1.3 SER Analysis Framework . . . 71
6.2 MBU Aware Voltage Scaling . . . 71
6.3 Power-Aware ID Optimization . . . 74
6.3.1 Interleaving Distance Power Model . . . 74
6.3.2 Optimal Interleaving Distance (OID) . . . 77
6.4 Experimental Results . . . 78
6.4.1 Interleaving Distance Effect on V_{tol} . . . 79
6.4.2 Power Reduction . . . 82
6.5 Conclusions . . . 83

7 Dynamic Voltage Scaling for SEU-Tolerance in Low-Power Memories 84
7.1 Soft-Error Immunity 86
7.1.1 Built-In Current Sensor (BICS) 86
7.1.2 Column-based V_{dd} Memory 87
7.2 Adaptive Soft-Error Tolerance 89
7.2.1 Adaptive Supply Voltage Strategy 89
7.2.2 Memory Timing Access 90
7.2.3 Minimum Recovery Voltage V_{high} 92
7.3 Probabilistic Power Model 96
7.3.1 Power Model using V_{high} and DRV 96
7.4 Experimental Setup 98
7.5 Experimental Results 98
7.5.1 Dynamic Noise Margin (DNM) for SEU Analysis 98
7.5.2 Power Reduction 100
7.6 Conclusions 103

8 Practical Issues with Fault Tolerant Low-power Memories 104
8.1 Measurement of Row/Column Leakage in Real Memory 104
8.2 Fuse Architecture and Programming 105
8.3 Generating Accurate Supply Voltages 107
8.4 Verification of SEUs/MBUs and SER 108

9 Conclusions and Future Work 109
9.1 Thesis Contributions 109
9.2 Future Work 111

Bibliography 112

List of Figures

1.1 Chain Diagram between Fault tolerance and Low-Power in Memory Design. . . . 3

2.1 Six-transistor SRAM cell (6T Cell) . . . 6
2.2 Cell size ratio (CR,PR) effect on node voltage with PTM model . . . 8
2.3 Static Noise Margin (SNM) of cell in VTC (Read, Write SNM) . . . 10
2.4 Optimal energy point is near sub-threshold and sub-threshold scaling increases the failure probability. . . . 12
2.5 Variance and ratio of DRV to supply voltage using 32nm, 45nm, 65nm, 90nm PTM model [4] . . . 14
2.6 Lognormal Distribution of 4K cells DRV at 45m PTM model . . . 15
2.7 Column based V_{dd} line for active power reduction in memory array . . . 16
2.8 Eight-transistor (8T) CMOS SRAM cell . . . 17

3.1 Single Event Upsets (SEU) due to the energy particles in Transistor . . . 21
3.2 Modeling the induced current pulse to simulate Soft Error effect in gate level . . . 21
3.3 Concept of Redundant Spare Rows and Columns . . . 24
3.4 Interleaving Distance (ID) Scheme for Soft Error avoidance . . . 25
3.5 Built-In Current Sensors (BICS) detect particle strikes by monitoring the virtual supply and ground. . . . 26

4.1 Voltage vs. SNM(Read) Using PR=0.5, 0.8, 1.0 and 2.0 . . . 29
4.2 Pull-up Ratio (PR) vs. $\text{SNM}_{product}$ . . . 31
4.3 Comparison of V_{thp} between Non-optimized and Our method: our method improves NBTI degradation of PMOS transistor compared to non-optimized cell. . . . 36

5.1 Sub-threshold V_{dd} effect on 1K cell's leakage and DRV distribution of 4K cell with PTM [4] model . . . 40
5.2 Monte Carlo Framework for DRV_{map} manipulation . . . 43
5.3 Cell DRV vs. Cell leakage at V_{dd}=130mV on 16K sample cells . . . 49
5.4 Our yield model matches simulation data for V_{stdby} calculation with good accuracy. . . . 52

5.5 V_{dd}, redundancies effect on yield with 1K SRAM 53
5.6 8T cells need a smaller voltage to have same read stability according to SNM. 56
5.7 Active operating voltage V_{act} distribution of 8T cells are significantly lower than 6T cells. 57
5.8 Memory architecture using the optimal voltage V_{stdby} and V_{act} for idle mode and active mode 58
5.9 Comparison with previous methods with 1K SRAM 62
5.10 Redundancy Effect on Leakage at Required Yields for 1K SRAM 63

6.1 Monte Carlo Framework for SER_{map} manipulation 70
6.2 V_{tol} and worst-case voltage over n=1000 SER_{maps} at a normal flux [94] . 73
6.3 Example of ID effects (ID=1,2 and 4) on power in SRAM 75
6.4 ID vs. V_{tol} and power optimal condition in 1K, 4K, 16K and 64K SRAMs with Flux1 79

7.1 Gate-level SEU simulation methodology are used to analyze circuit robustness. 85
7.2 Previous works have separately used Built-In Current Sensor (BICS) for error detection and Column-based V_{dd} Array for dynamic power savings depending on the operation (read or write). 88
7.3 Our approach uses Built-In Current Sensor (BICS) together with a Column-based V_{dd} Array to detect SEUs at a column granularity. 89
7.4 A Monte Carlo framework is used to analyze the timing and power of the low and high supply voltage levels. 91
7.5 Memory worst case delay is fit to a non-linear model for various array sizes. 91
7.6 Plot (Column size N vs. $t_{recover}$) in different I_{peak}. Recovery time of memory array increases linearly as column size N increases. 93
7.7 Simulation results using our feedback (BICS and dual V_{dd}) on a 1024 bit memory (32 cells in a column). Case (A): SEU flips memory cell, Case (B): SEU but cell recovers due to a higher voltage. 94
7.8 Calculation of V_{high} lower-bound using t_{worst} model and $t_{recover}$ simulation with I_{peak}=*3.25E-05* shows that the criterion is satisfied around 0.9V in 1024K SRAM 95
7.9 Peak current's amplitude (I_{peak}) vs. $t_{recover}$ in different dual V_{dd} combinations (1K SRAM) V_{high} determines the memory tolerance to a given I_{peak} amplitude and it should be calculated to optimal V_{dd} level to reduce the power. 99

8.1 Electrical Fuse (eFuse) structure that is programmed by applying a high voltage during NMOS activation. 106

List of Tables

4.1 Cell Layout Area Overhead using PR Value. 32
4.2 Area Overhead of SRAM including Arrays and Peripherals. 32
4.3 Performance (Delay$_{write}$) Comparison to Original SRAM. 33
4.4 Power Reduction @ 25,50,75 and 100°C using Proposed Methods on a 6T Cell. 34
4.5 SNM Variation Comparison between Previous and Our Method. 35

5.1 Comparison of Leakage at 99.9% Yield when (6,6) redundancies are available. (6T cell) 65
5.2 Comparison of Leakage at 99.9% yield when (6,6) redundancies are available. (8T cell) 65

6.1 Comparison of Power Reduction Effect with various Radiations (flux1, flux2 and flux4) in various SRAM sizes (W=8) 81

7.1 Power Reduction Results when Radiation strikes memory Once in various Sizes ($p = 0.1$) 102
7.2 Power Reduction Results when Radiation strikes memory Twice in various Sizes ($p = 0.2$) 102

Abstract

Low-Power Methodology for Fault Tolerant Nanoscale Memory Design

by

Seokjoong Kim

Millions of mobile devices are being activated and used every single day. For such devices, energy efficient operation is very important; low-power operation enables not only long battery time but also improves energy efficiency of the servers that communicate with the mobile devices. However, reduced noise margin due to low-power operation and process variation due to nano-scale transistor feature sizes increase the number of errors in both mobile and server devices. Thus, low-power issues and reliability are strongly related.

This work focuses on reliable, low-power methodologies for SRAM memories. It is the first to consider SRAM cell optimization for power and reliability simultaneously. The main contributions are the following. To guide parametric hard faults, this work addresses energy optimality and yield considering redundant spare rows and columns. This work also describes a method for soft error tolerant low-power memory design using an architectural technique to avoid Multiple Bit Upset (MBU) at low voltages. Then methods using dynamic voltage scaling for soft error tolerant low-power memory designs are investigated.

This thesis results in the improvement of memory power consumption and increases the reliability of memory arrays. Using cell optimization, redundancy utilization, interleaving techniques, and adaptive dynamic voltage scaling, memory reliability is improved and power reduction is reduced by 10%-40% depending on the method applied without sacrificing error tolerance.

Dedicated to my lovely family,

Jeena and Dan.

Acknowledgments

I would like to thank my advisor Prof. Matthew Guthaus for his invaluable guidance throughout the course of my PhD. His insights and encouragement have been a great source of inspiration for this work. I am also grateful to Prof. Joel Ferguson, Prof. Jose Renau and Dr. Krishnan Sundaresan for agreeing to be a part of reading committee.

The many hours I have spent at work have been very stimulating and enriching. I am proud to have been a part of VLSI DA Lab due to a great research environment and wonderful colleagues who I could interact with. I would especially like to thank Xuchu Hu, Sheldon Logan, Rajsaktish Sankaranarayanan, Marcelo Siero, Jas Condley, Chasen Peters, Derek Chan and Keven Woo. Also, I thank Mohammed Jamil for support of my thesis work after I joined Oracle.

Last, but not the least, thanks to God and my family, my parents and parents-in-law for their love, encouragement and support. I thank my lovely wife and sweet son, Jeena and Dan for their love and patience during the course of my PhD.

Chapter 1

Introduction

The recent trend of increasing energy costs and usage of mobile applications has sparked a strong interest in energy efficient systems that use non-traditional energy sources such as battery, solar, scavenging, etc. Many of these alternative energy sources provide very small amounts of power in the μW range and therefore cannot be used with traditional super-threshold ($V_{dd} > V_{th}$[1]) circuits. To address this, previous researchers have shown that memory can operate with supply voltages well into the sub-threshold region ($V_{dd} < V_{th}$) [12, 28, 81] and that this region is even energy optimal [81].

Among the methods for power reduction, the mainstream method is lowering the supply voltage [13, 63, 80, 82, 84]. Scaling the operating voltage can reduce the power consumption very effectively in super-threshold region ($V_{dd} > V_{th}$). Using this voltage scaling, battery powered device can outlast their operation time while keeping the minimum performance compared to the device without voltage scaling. The advantage of voltage scaling method also apply in sub-threshold region ($V_{dd} < V_{th}$). However, in sub-threshold

[1] Voltage at which channel formation occurs and it conducts source and drain of a transistor.

region, some reliability issues occur due to process variation and external noise.

Increased process variation and unknown external noise cause reliability problems in the ultra-low-power circuits. In the low V_{dd} region, circuits can be easily affected by noise due to the low V_{dd}'s weak driving strength, and it is hard to expect number of errors and location of errors due to the random process variation. The cause of these reliability problems can be either permanent (hard) faults [45] or transient (single event) faults [20, 55, 57, 95]. Often hard faults during the manufacturing process due to process variation and affect the manufacturing yield. Other type of faults, transient faults, mostly occur when the external noise/energy particle hits the circuit's operation node such as memory cell-flip[2] to malfunction due to an atomic energy particle spike. The rate of both permanent and transient faults increases in low-power circuits and must be a primary consideration in real applications.

1.1 Fault Tolerant Low-Power Memory

At low V_{dd}, V_{th} mismatch and noise can affect memory power, performance and reliability so that it should be considered at early design stages. Figure 1.1 describes the chain diagram of fault tolerant low-power memory in a clockwise rotation. The need for low-power results in the various low-power techniques, which reduce the feature size and worsen the process variation. As a consequence, the reduced V_{dd} and process variation bring the reliability issues such as errors in memory. Specifically, hard-errors (permanent faults) and soft-errors (temporal faults) happen in memory. To deal with these, researchers focus on fault tolerant techniques such as increased supply voltages and fault-aware circuit

[2] Stored logic value is inverted in memory cell due to the external energy.

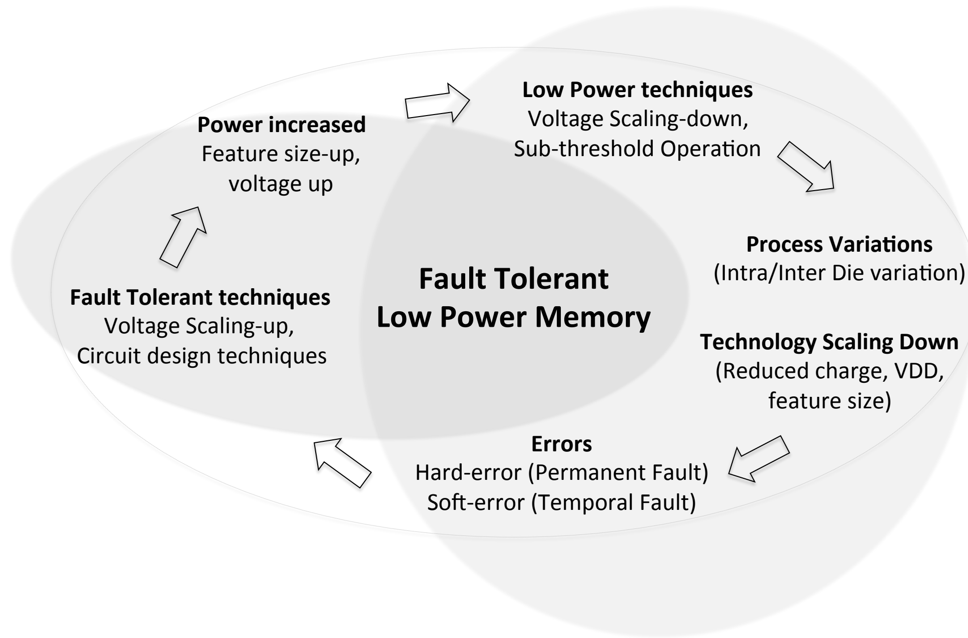


Figure 1.1: Chain Diagram between Fault tolerance and Low-Power in Memory Design.

design. It means that the feature sized-up and V_{dd} increased eventually.

Through this chain, we can notice that low-power issues are not an independent problem. All low-power issues are connected to the correspond reliability issues and vice versa. This means that both (low-power and reliability methodologies) should be considered simultaneously. This work describes methodologies considering low-power and fault tolerant issues, simultaneously.

1.2 Thesis Contributions and Outline

This thesis describes proposed methodologies for combined low power and reliability. Each proposed method contributes to the low-power improvement as well as the fault

tolerance. Specifically, our major contributions are as follows:

- For fault tolerant low-power memory cell design, transistor sizing method considering reliability is proposed [43].
- For permanent error free design, a memory redundancy utilization method considering leakage reduction is proposed [41].
- For static soft error tolerant design, a memory interleaving distance optimization method considering Multiple Bit Upsets (MBU) is proposed [42].
- For dynamic soft error tolerant design, dynamic V_{dd} scaling method using Built-in Current Sensor (BICS) is proposed.

All these contributions reduce the power consumption effectively and achieve fault tolerance. Using cell optimization, redundancy utilization, interleaving techniques, and dynamic voltage scaling, the power improvement and reliability of memory increased by 10%-40% depending on the error types without sacrificing yield and soft error tolerance.

The rest of this thesis proceeds as follows: Basic features of low-power fault tolerant memory are reviewed in Chapter 2 and Chapter 3. For low-power tolerant memory design, a memory cell optimization technique considering noise margin is described in Chapter 4, redundancy utilization method is proposed in Chapter 5, a memory interleaving distance optimization method is proposed in Chapter 6 and dynamic V_{dd} scaling method using BICS adaptive feedback system is proposed in Chapter 7. Each Chapter shows separate results and Chapter 9 concludes the thesis and motivates future works.

Chapter 2

Low-Power Memory Design

With technology scaling, device variability in nanometer SRAM cell design is increasing. The wider spread of local mismatch leads to reduced SRAM reliability. To estimate the robustness of memory cell, Static Noise Margin (SNM) has become one of the major concerns for SRAM design. In Section 2.1, basic SRAM structure/operation and the definition of SNM is described in detail. Then Section 2.2 illustrates the methodologies for low-power memory design.

2.1 Static Random Access Memory (SRAM)

2.1.1 Memory Cell Structure (6T Cell)

In Figure 2.1, a typical 6T CMOS SRAM cell is shown with two NMOS access transistors (N3 and N4) and four storage transistors (N1, N2, P1 and P2) that form a cross-coupled inverter. The 6T-cell design must allow non-destructive read operations and reliable write operations.

Figure 2.1: Six-transistor SRAM cell (6T Cell)

2.1.2 Read/Write Operation

The read operation is started by enabling the WL which connects the precharged bit lines to the internal node of the cell. To read a 1, the bit line voltage V_{BL} remains at a precharge level and the complementary bit line voltage V_{BLB} is discharged through two transistors, N1 and N3, in series. The sizing of N1 and N3 should maintain the node voltage on I1 so that it doesn't cause a switching of the other inverter's input (P2 and N2). If the sizing violates this safety voltage level, it will discharge I2 and result in a destructive read. The maximum voltage ΔV which is allowed without flipping the cell during a read access can be expressed as [64]:

$$\Delta V = \frac{1}{CR}\Big(V_{DSATn} + CR(V_{DD} - V_{THn}) - \sqrt{V_{DSATn}^2(1+CR) + CR^2(V_{DD} - V_{THn})^2}\Big) \tag{2.1}$$

where CR (often called β) is the cell ratio defined as:

$$CR = \frac{W_{N1}/L_{N1}}{W_{N3}/L_{N3}} = \frac{W_{N2}/L_{N2}}{W_{N4}/L_{N4}} \tag{2.2}$$

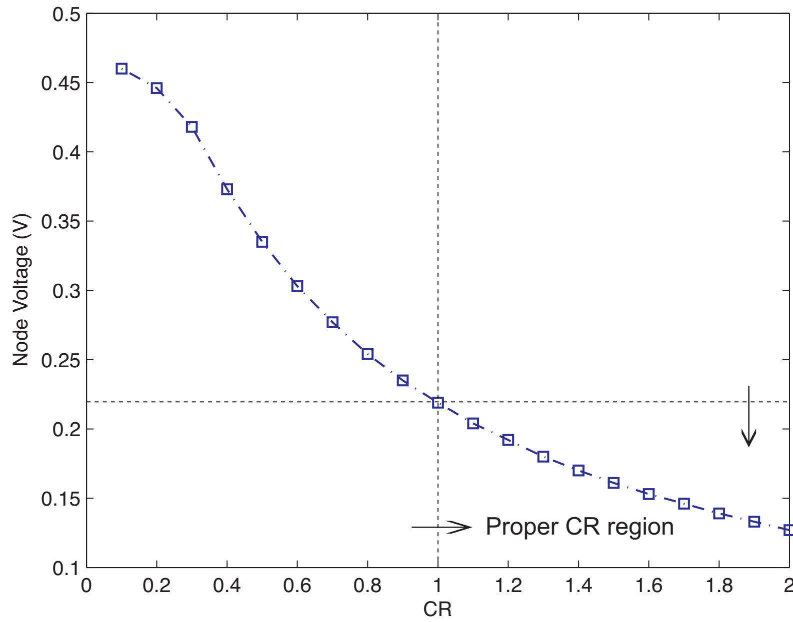
To demonstrate the transistor's size effect on node voltage, we simulated SRAM cell with 45nm PTM [4] model parameter as Figure 2.2 shows. The dependence of ΔV on CR is shown in Figure 2.2(a). In order to ensure the stable read access, CR should be large enough that ΔV is less than V_{thp}. Typically, in order to perform a non-destructive read, CR must be greater than one. Adding an adequate noise margin and depending on the target application, the CR can range from approximately 1-2 as in Figure 2.2(a).

During write operations, one of the bit lines, BL or BLB, is driven to the precharged level V_{DD} and the other is driven low. When the word line is enabled, the bitline and the internal node are connected and the voltage levels overwrite the internal nodes. If the transistors P2 and N4 are properly sized, for example, then the value on node I2 will be overwritten successfully. The node voltage derived from the write operation can be expressed as [64]:

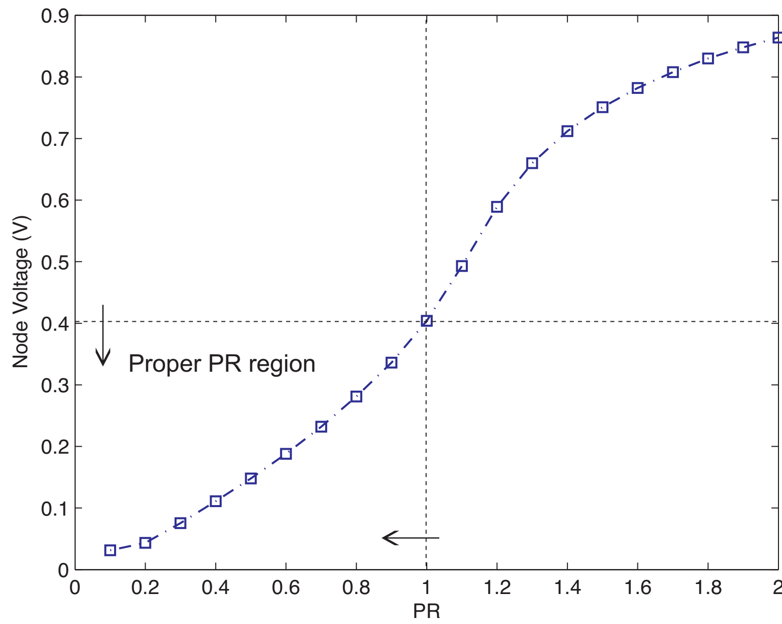
$$V = V_{DD} - V_{THn} - \sqrt{(V_{DD} - V_{THn})^2 - 2\frac{\mu_p}{\mu_n} PR \cdot K} \tag{2.3}$$

where

$$K = (V_{DD} - |V_{THp}|)V_{DSATp} - \frac{V_{DSATp}^2}{2} \tag{2.4}$$



(a) CR vs. Node voltage



(b) PR vs. Node voltage

Figure 2.2: Cell size ratio (CR,PR) effect on node voltage with PTM model

and PR is the pull-up ratio of the cell defined as:

$$PR = \frac{W_{P1}/L_{P1}}{W_{N3}/L_{N3}} = \frac{W_{P2}/L_{P2}}{W_{N4}/L_{N4}} \tag{2.5}$$

If P2 and N4 are not properly sized, the cell's voltage level will not be flipped and a write failure occurs. For example, assume that BLB is precharged to V_{DD} and the node voltage I1 is initially set to zero voltage. If WL is enabled, the precharged BLB will change the voltage level of node I1 to near V_{DD} level. But if P2 is over-sized, the gate voltage level of N1 will exceed the threshold voltage, V_{thn}, and discharge I1. As shown in Figure 2.2(b), the PR value should be selected to make ΔV smaller than the threshold voltage V_{thn} to ensure a stable write operation.

Besides the read/write operation, SRAM spends most of the time in idle mode. This mode is called stand-by/hold operation. In stand-by operation, the wordline WL turns off the access transistor (N3, N4) and keep the stored value. Since this operation is critical in low-power SRAM design, it will be illustrated further in Section 2.2.2.

2.1.3 Static Noise Margin (SNM)

SNM is widely used as the criteria of stability. SNM is one of the most important factors in analyzing SRAM cell design. As technology scales from 250nm to 45nm, V_{th} mismatch increases and the SRAM SNM is reduced up to four times [87].

The traditional butterfly SNM approach using the input voltage to output voltage transfer curves (VTC) is the most popular one. Figure 2.3 shows the read SNM and write SNM in relation to the cross-coupled inverter characteristics [71,83]. Read SNM is defined

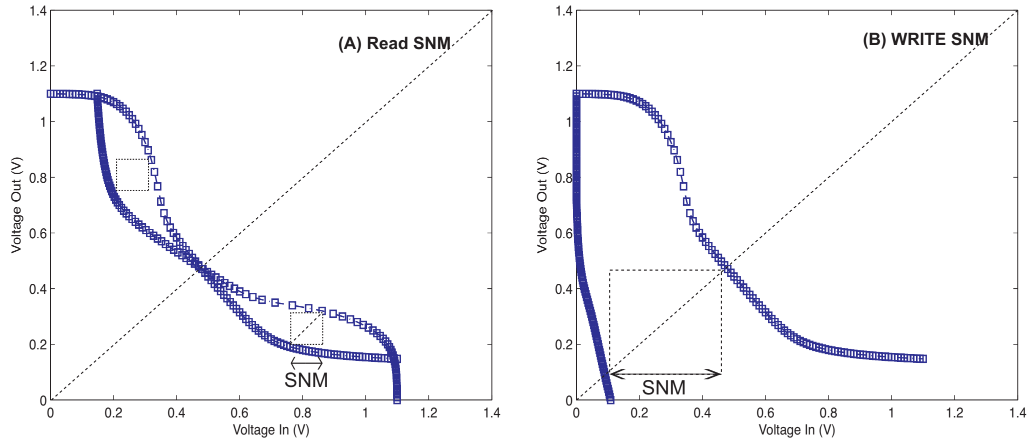


Figure 2.3: Static Noise Margin (SNM) of cell in VTC (Read, Write SNM)

as the width of embedded maximum square in VTC curve and write SNM is defined as the width of the smallest square that can be embedded between the lower-right half of VTC curve.

2.2 Methodologies for Low-Power Memory

2.2.1 Current State of the Art: Toward Sub-Threshold SRAM

Both resilience of memory and the drastic scaling of supply voltages into the sub-threshold region have independently received significant interest. Without improved reliability of low-power devices, it is not possible to exploit their energy efficiency in any practical system that requires reasonable reliability. This section summarizes the recent work targeting sub-threshold memories.

In recent years, there has been a significant amount of work on circuits and design for sub-threshold circuits [12, 28, 66, 74, 81, 89, 90]. Wang et al. demonstrated that the

optimal energy point for an FFT processor was actually in the sub-threshold region due to high-switching activities [81]. The subsequent works have focused on sub-threshold memories [12, 89], effects of technology scaling on sub-threshold circuits [28, 66], new circuit techniques [49, 73] and sub-threshold optimization [29].

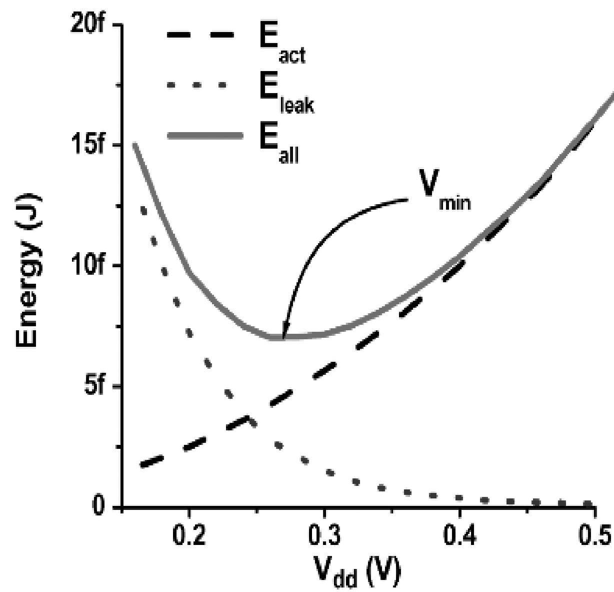
In prior works [89], the energy optimal voltage was selected to minimize a combination of active and leakage current as shown in Figure 2.4(a). The energy optimal point depends on the circuit activity rates, capacitance, and frequency.

These previous works, however, only cursorily investigated the effects of such supply scaling on the reliability of such circuits by measuring errors on fixed architectures and circuit styles. For example, in [28], the technology impact on bit error rates is shown to be a problem as illustrated in Figure 2.4(b). In [89], bit error rates for read, write and hold are analyzed for a specific SRAM with a new circuit style at various supply voltages.

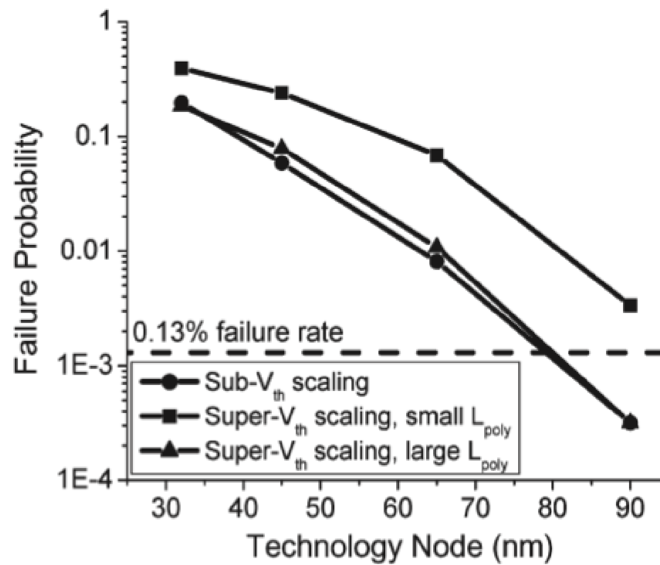
2.2.2 Voltage Scaling techniques

Voltage scaling can make SRAMs consume less power. The key idea is to control the voltage bias according to the cell operating mode. Normally, SRAM cells spend most of the time in idle mode. During the idle mode, the WL is deasserted and the internal node voltage of the cell maintains the previously stored value. The supply voltage for the SRAM during idle mode can be reduced as long as the previous stored value is kept.

To keep the correct data with the minimum energy, people use the minimum hold voltage in idle mode so that the memory cell can hold the correct logic value with the minimum voltage [11, 82]. This small amount of voltage to hold the correct value is called Data Retention Voltage (DRV). Normally DRV is less than the threshold voltage and can be



(a) Optimal energy point is near sub-threshold (V_{dd} ¡ V_{th}) region (inverter chain) [89]



(b) Voltage scaling increases read failure probability for a single SRAM cell at 250mV under different scaling strategies. [28]

Figure 2.4: Optimal energy point is near sub-threshold and sub-threshold scaling increases the failure probability.

calculated using the static noise margin calculation based on DC simulation when SNM is almost equal to zero.

Using the full voltage in active mode and DRV in idle mode can be a very effective approach. It reduces the leakage power of the memory array efficiently without any performance degradation while the memory operates in active mode. Since most of the cells, except those in the accessed row, are normally in the standby mode, the leakage power is minimized with a low DRV. Previous works show that the DRV limit is less than 80mV for a 45nm, 150mV for a 90nm PTM technology at 60 °C [82]. However, using the minimum DRV level in a memory array makes the number of faults (Hard Faults and Soft Errors) increase dramatically because DRV is very sensitive to the process variation (V_{th} variation). As the technology scales to the 65nm and 45nm regions, both V_{dd} and V_{th} decrease so that the DRV variation dramatically increases. We observe using the Predictive Technology Models (PTM) [4] that both the variation and the portion of DRV are increasing in Figure 2.5.

Researchers often analyze DRV and use a guard band voltage. In Figure 2.6, the DRV has the shape of the lognormal distribution. It has a long tail in the worst case DRV direction. This means two things: a) a small portion of DRV determines the worst case DRV and b) using the worst DRV may not promise reliable hold operation in idle mode due to the variation in low V_{dd} regions. In other words, to optimize the power, the worst case DRV should be minimized, and to assure the reliable idle operation, a guard band voltage above the worst case DRV is required. As expected, this is a conflict. To resolve this, we propose one method in Chapter 5.

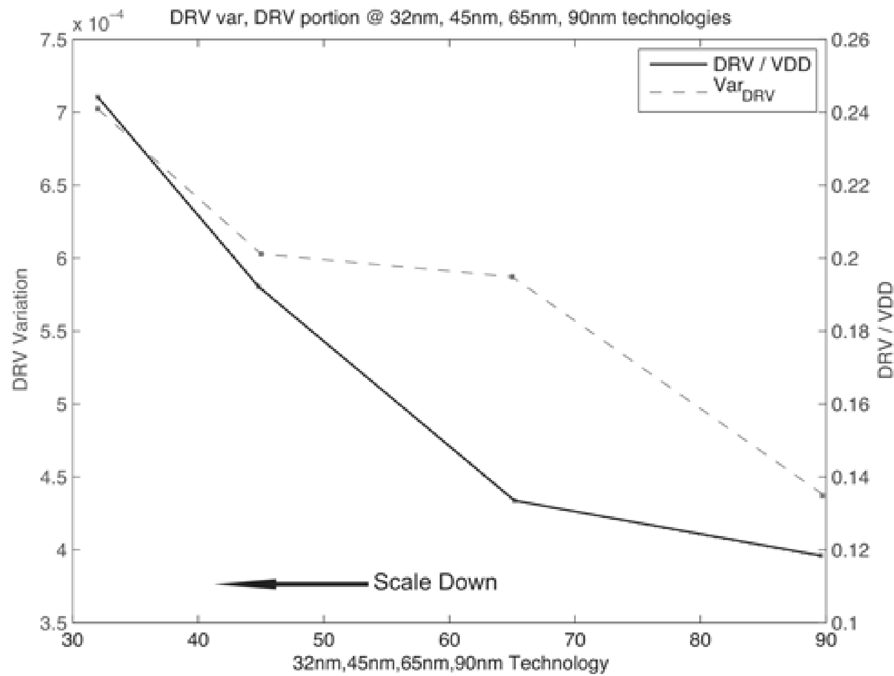


Figure 2.5: Variance and ratio of DRV to supply voltage using 32nm, 45nm, 65nm, 90nm PTM model [4]

2.2.3 Transistor/Circuit level techniques

One way in which designers have tried to address the power consumption problem is by applying reverse/forward body-biased SRAM cache and high-V_{th} transistors [38]. Using high-V_{th} for suppressing the leakage and forward body biasing for the performance is efficient to reduce the power consumption in SRAMs [38] but this method needs additional fabrication steps and masks for super halo doping to use high V_{th} transistors.

In similar motivation, people used the transistor gate sizing method [23] for the power reduction. This method lowers I_{ds} current by reducing the ratio of transistor width and length (W/L). For instance, if the designer knows the critical path and can list the transistors that belong to the non-critical path, this method modulates the non-critical path's

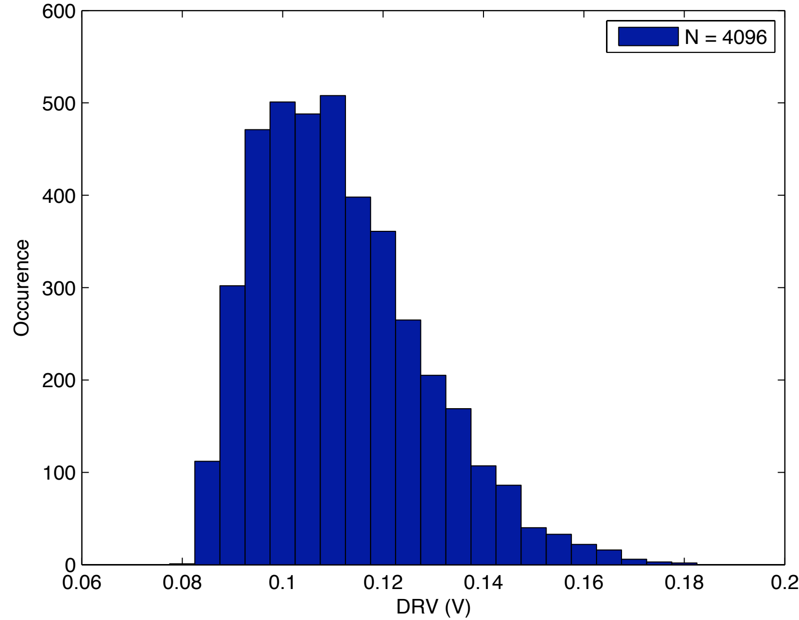


Figure 2.6: Lognormal Distribution of 4K cells DRV at 45m PTM model

transistor gate size (the transistor's W/L) to reduce drain-source current I_{ds} within the bound that the critical path delay is not violated. The advantage of this method is reducing the overall power with transistor technique without any additional fabrication process.

Using a sleep transistor is one of the common strategy to suppress the leakage power. This method is also called power-gating [14, 34, 37] and uses the connectivity of the sleep transistor as a switch depending on the operation. Since the sleep transistor is located in the charging or discharging path of a circuit, the sleep transistor size and location is carefully selected depending on the current charging/discharging pattern and the amount of current flow to minimize the impact on the power-gated circuit's timing. Similarly, the gated-supply voltage (V_{dd}) technique [36, 61] and gated-ground technique [1, 2] use a transistor as a switch to reduce the power depending on the circuit operation.

Column-based V_{dd} method is an architectural technique proposed recently [27, 92]. Figure 2.7 shows the overview of this method. Each cell's power line V_{dd} is connected

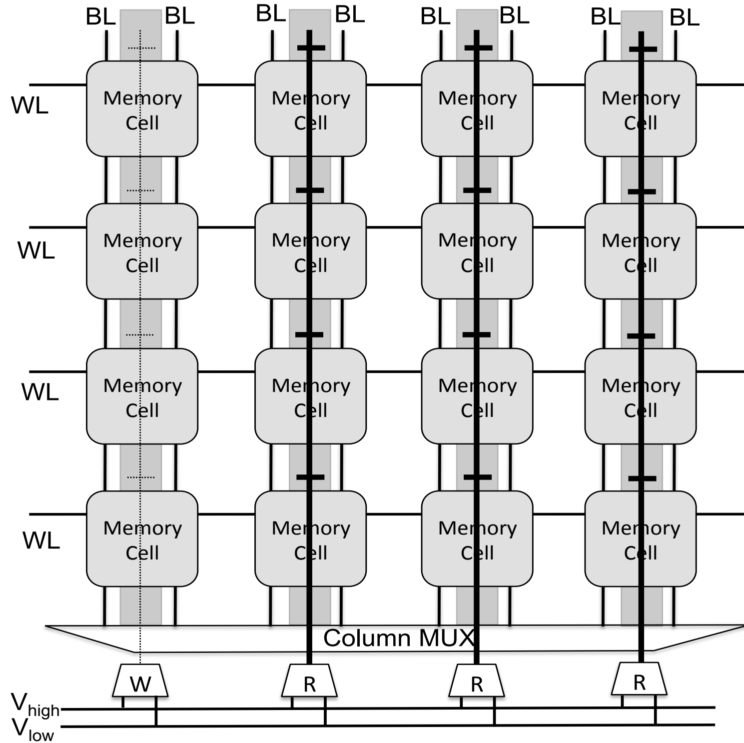


Figure 2.7: Column based V_{dd} line for active power reduction in memory array

to the global power line in columns. This idea is based on the fact that SRAM read operations need a higher V_{dd} compared to the write operation for valid operation. It has an advantage of power reduction using two separate voltages (V_{high} and V_{low}) depending on the read/write operation.

Besides this, many design techniques have been proposed to reduce the subthreshold leakage, including drowsy caches [21, 40], asymmetric-cell [26].

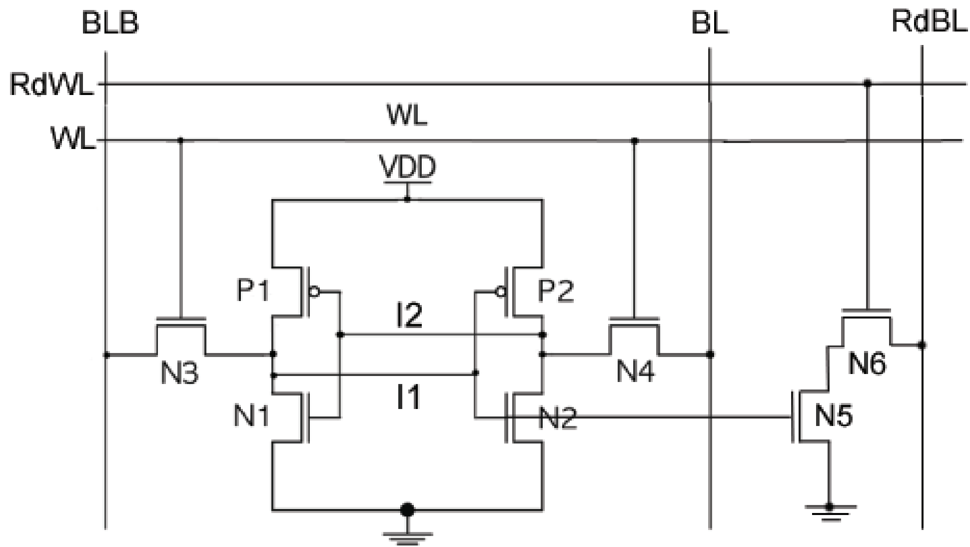


Figure 2.8: Eight-transistor (8T) CMOS SRAM cell

2.2.4 Cell Architecture Technique (8T Cell)

Using a different memory structure (8T Cell) is one of the efficient method for sub-threshold SRAMs. This method uses a separated read operation path and write operation path to improve read operation at low V_{dd}. As expected, power and robustness are improved compare to the traditional 6T cell. results in robustness and power improvement.

The fundamental stability problem in 6T cells is the stability of the read operation. Adding two transistors to a 6T cell can isolate the internal node during a read operation as shown in Figure 2.8 [11–13]. This, however, requires separate read and write word lines, RdWL and RdBL, for each cell. During a read operation, RdBL is precharged to V_{dd} and RdWL is enabled to turn on the transistor N6. Then the current from precharged RdBL moves through the transistor N6. Consider a read operation of the P2-N2 stored node value

in Figure 2.8. If the stored value of internal node I1 is V_{dd} level, this will turn on the transistor N5. Consequently, the current will be discharged without affecting the internal node I1 and I2 voltages. This completes a non-destructive read operation in 8T SRAM cell. Hence, the 8T SRAM cell can improve the following aspects of the SRAM cell: 1) the cell current I_{cell}, 2) the dynamic and leakage power, and 3) read stability. Since there is no disturbance during read and write operations, the read stability for an 8T cell has significantly larger SNM, we will compare our results to this in Chapter 5.

Chapter 3

Fault Tolerant Memory Design

As shown in Figure 1.1, low voltage operation and reduced transistor sizes increase process variation and increase errors in low voltage regions. In Section 3.1, the types of errors are introduced in detail and in Section 3.2, previous methodologies for tolerant design are summarized.

3.1 Type of Errors (Hard Faults and Soft-Errors)

Errors occurred in memory design are largely categorized to two types. One is hard faults and the other is soft-errors. Hard faults are described in Section 3.1.1 and soft-errors are introduced in Section 3.1.2, respectively.

3.1.1 Hard Faults (Permanent Errors)

Hard faults are called permanent faults because this type of faults can be detected and only repaired with repair techniques after the fabrication process. So it is very directly related to manufacturing yield.

The original works in the area of memory reliability focused on manufacturing hard defects [16]. The author showed that having square memory arrays with small amounts of redundancy or if it is not square having redundancy in the larger dimension is most beneficial for yield. Subsequently, Shi and Fuchs [72] provided probabilistic algorithms for reconfiguration of memories using spare columns and rows. In fabrication process, reliability engineers check the test structures using voltage-timing plot such as the Shmoo plot [6] and low voltage test [59] before mass fabrication with the initial post silicon data. However, realistically, in many cases, memory arrays can fail to read, write and hold the correct value even though the memory passed these tests in low voltage regions due to the increased process variation.

Several researchers have presented studies on static and dynamic SRAM cell stability [3, 9, 31, 91] while others have proposed self-repairing or adaptive body bias techniques to improve yield or power [3, 24, 51–53]. However, all of this reliable memory research has focused on the super-threshold region of operation. The few works on subthreshold memories [13, 28, 89] have focused on absolute energy reduction with only a cursory measurement of SNM and bit-error rates for a given design or circuit style.

3.1.2 Soft-Errors (Single Event Upsets)

SEUs are caused by alpha particles or cosmic rays that, when they collide with a surface, create temporary electron-hole pairs. In the past, these were common only in high-altitude (space) applications, but they are becoming more significant as process geometries and supply voltages shrink. Figure 3.1 shows the case that an energy particle such as alpha, neutron hits the channel of transistor and create the pairs of electron and hole. This means

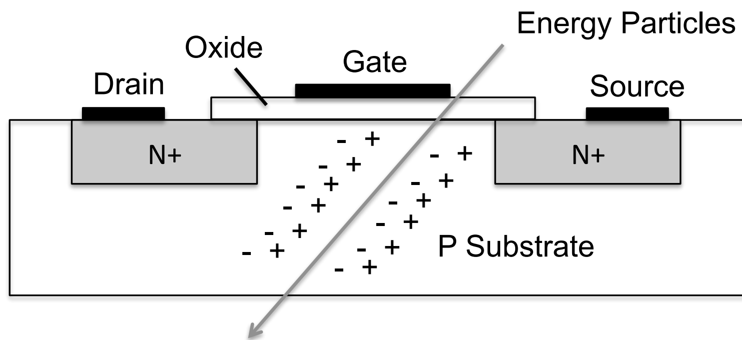


Figure 3.1: Single Event Upsets (SEU) due to the energy particles in Transistor

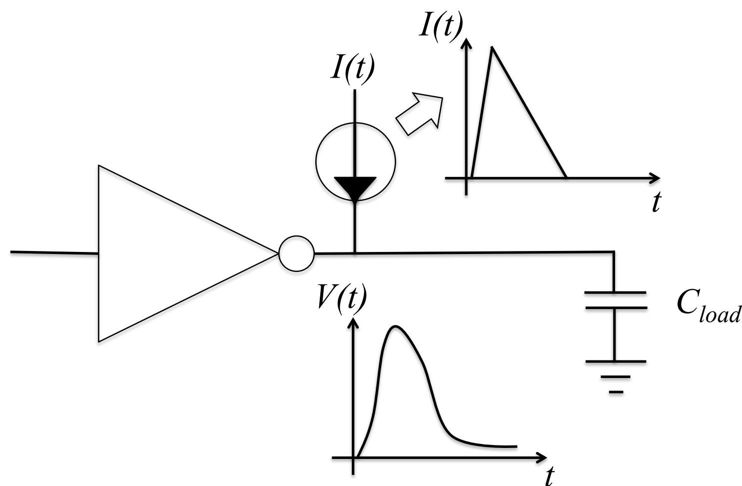


Figure 3.2: Modeling the induced current pulse to simulate Soft Error effect in gate level

that the transistor's characteristic such as I_{ds} changes depending on the amount of energy particles.

To simulate this effect, people use artificial pulse generated by the induced spike in Figure 3.2, attach the artificial pulse to the end of the node and simulate to see the effect of the induced energy particles at the transistor/gate level. Based on this simulation, previous works have proposed methods to calculate the upset rate [54, 65, 93]. These methods typically calculate the critical charge [22] needed to hold the correct value on a circuit

node. If radiation induces a higher charge than the critical charge, a transient error will happen. Since transient errors are not permanent, it is possible to recover from such faults, otherwise, there would be a soft error.

In the past, soft errors have been evaluated as Single Event Upsets (SEU) [57, 95] because the transient errors happened sparsely over the manufactured chip. As the transistor sizes scale into the nanometer regime and the supply voltage is lowered, however, multiple simultaneous errors have begun to appear. These multiple simultaneous errors are called Multiple-Bit Upsets (MBUs) [20, 55] and are a significant issue in modern designs.

In memory devices, this pulse directly strikes the storing node and affects the node voltage depending the energy particle's strength. However, in combinational logic devices, the pulse is only captured when the induced pulse is propagated to and captured by a latch.

SEUs are typically prevented by electrical, logical, and temporal masking. Electrical masking is the circuit robustness to added electron-hole pair charge; logical masking is the prevention of a fault from propagating to the state or primary outputs; and temporal masking is when a fault misses the timing window of a flip-flop or latch. Specifically, in low-power memories, the vulnerability to SEUs increases tremendously due to:

1. Decreased CV nodal charge allowing fewer electron-hole pairs to upset a node.
2. Decreased static noise margins (SNM) allowing a smaller transient pulse to flip.
3. Decreased attenuation due to small I_{on} currents delaying recovery from an SEU.

This means that SEUs in low-power memories and logic drastically increases even in non-space applications.

3.2 Methodologies for Fault Tolerance

In the non-sub-threshold region, there have been many works on reliable, low-power memories. Memories, due to their small device sizes and large numbers of devices, are typically the most sensitive to manufacturing and SEU errors. Although it is not feasible to perform a complete analysis of all memory research, the most pertinent works are summarized.

3.2.1 Redundant Row/Column for Repair Hard Faults

The testing community has developed methods to repair manufacturing faults using redundant rows and columns to improve yield. Redundancy is effective approach to increase the yield by replacing the faulty rows and columns with spares [16, 24, 50, 51, 72]. Applying row or column redundancy inevitably increases the power consumption due to the spare rows and columns, however, more importantly, it can improve the yield. For example, Figure 3.3 shows a 64-bit memory array with a small number of row and column redundancies.

However, it is a well-known NP-complete problem to find an optimal coverage with a limited set of row/column redundancies. One of the previous methods [44] employed a reduction method with an exhaustive search that gives the exact solution, but it is slow when the number of faults is large and may not find a feasible solution even if one exists. Other more recent prior works have focused on redundancy analysis (RA) heuristics for built-in-self-repair (BISR) such as Repair Most (RM) and CRESTA [35]. RM is a fast greedy approach, but does not guarantee the global optimal solution. CRESTA enumerates

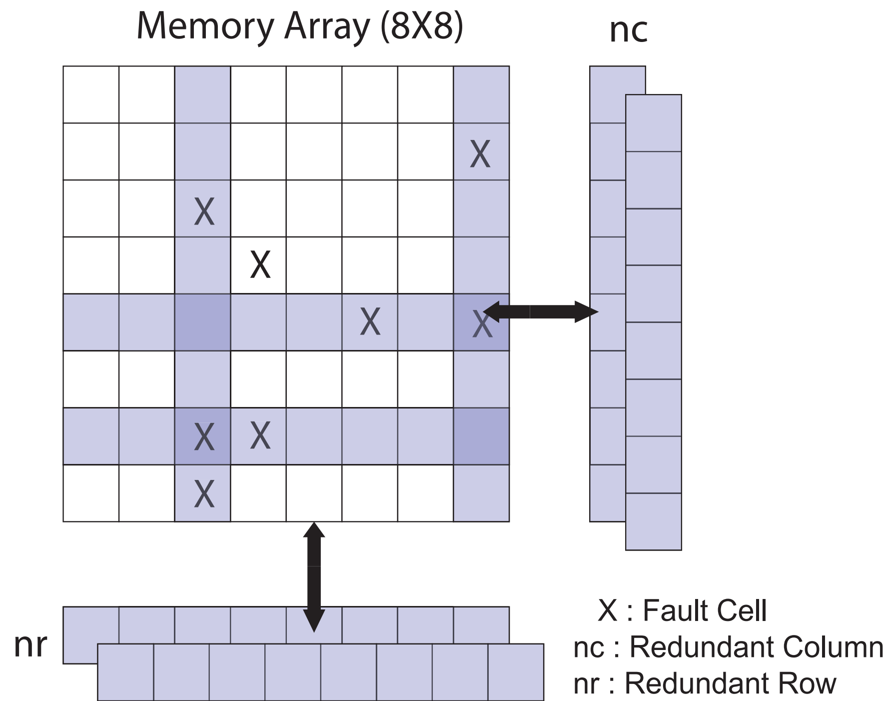


Figure 3.3: Concept of Redundant Spare Rows and Columns

all possible cases to give an exact solution, so the hardware cost for CRESTA is very high on large problems. More recently, another approach called IntelligentSolve [58] was proposed using depth-first-search (DFS) for use during on-line fault detection. These previous works have focused entirely on super-threshold memory optimization of hard manufacturing faults.

Besides redundancy to repair parametric hard faults, several researchers have presented studies on static and dynamic cell stability [3, 9, 31, 91]. Several researchers proposed self-repairing or adaptive (e.g. body bias) techniques to improve yield or power [3, 24, 51–53] and yet others have focused on ECC in caches [39].

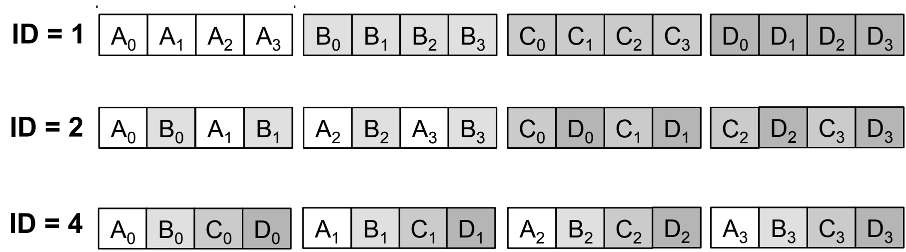


Figure 3.4: Interleaving Distance (ID) Scheme for Soft Error avoidance

3.2.2 Various Techniques (ID, ECC and BICS) for Soft Errors

To reduce the soft error rate, many previous works employ architectural techniques such as Error Correcting Codes (ECC) [25]. Error Correction Codes (ECC) add additional parity bits to original data bits to detect/correct errors. ECC can detect soft errors depending on the the number of parity bits. Single Error Correction Double Error Detection (SECDED) scheme is normally used for ECC due to its simple architecture, but Double Error Correction (DEC) can be implemented using more logics and gates and increases power.

Due to the overhead, ECC alone is not a reasonable solution especially for MBU. Many reliability engineers combine several techniques to increase MBU tolerance. In memory array design, designers employ an interleaving scheme to avoid MBU. The interleaving distance (ID) is the physical distance between bits that belong to same word. One previous work [5] described the interleaving distance (ID) effect on the memory failure rate with an analytical error model. They showed that memory having the maximum ID is beneficial for soft error immunity as the portion of MBUs increase. However, they didn't consider this effect on power consumption. Figure 3.4 shows the simple example of ID usage. For maximum fault tolerant design, the maximum ID is preferred, for example, ID=4 case.

To detect soft errors dynamically while the memory operates, researchers have

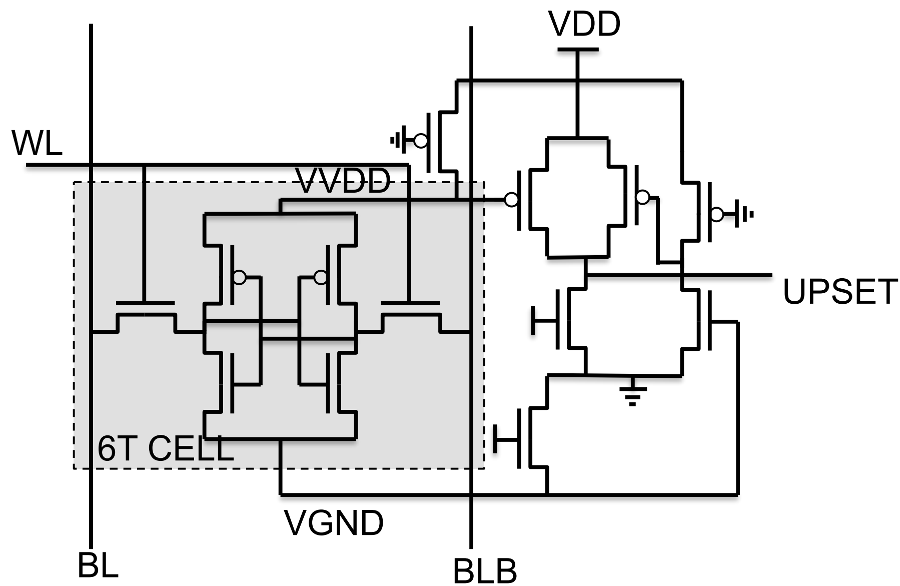


Figure 3.5: Built-In Current Sensors (BICS) detect particle strikes by monitoring the virtual supply and ground.

added Built In Current Sensors (BICS) which connect to V_{dd} or V_{ss} [56, 67]. The basic idea is that if the energy particle hits the memory cell, the BICS detects V_{dd} or V_{ss} line fluctuation. However, the area and power are increased due to the additional sensors located to each column.

Figure 7.2(a) shows a BICS implemented alongside a 6T SRAM cell [56, 67, 78]. The BICS connects to each column at the bottom of the array. When a particle strikes an internal node of any memory cell in the column, the voltage of the internal node fluctuates due to the electron-hole pairs and immediately decreases the virtual V_{dd} (*VVDD*) of the BICS. This turns on the PMOS transistor in pull-up path of the BICS which asserts the *UPSET* signal to indicate the presence of a transient particle.

Circuit level techniques can increase the soft error immunity using hardened memory cells [8] and/or voltage scaling [15, 85]. The basic idea of hardened memory cell

is increasing a capacitance of stored node to increase the critical charge Q_{crit} level. This method improves the soft error tolerance but it affects to the memory performance due to the increased capacitance.

3.2.3 Dynamic Noise Margin (DNM)

Dynamic Noise Margin (DNM) [19] can also estimate SRAM cell's dynamic stability for the transient SRAM behavior. Recently, many previous works proposed different analysis methods [32, 79, 83, 91] to identify DNM. DNM quantifies a memory cell's fault tolerance with the transient voltage state, not static voltage condition. It means that DNM can quantify the tolerability of memory cell to the external noise and energy particle such as glitches, alpha and neutron particles causing SEUs that are not instantaneous. It is hard to characterize analytically the dynamic behavior due to the latched style structure of SRAM cell, the different noise amplitude and the different noise width depending on the altitude and location. Chapter 7 analyzes the DNM in detail.

Chapter 4

SNM-Aware Cell Optimization of 6T SRAMs

In this chapter, we present our proposed method, which is SNM-aware cell optimization based on 6T cell structure to lower SRAM power and considering reliability. Our goal is to select a proper Pull-up transistor's Ratio (PR) value that can guarantee reliability, robustness and power reduction including both active power and stand-by power. Without an upperbound in PR value selection, the large pull-up transistor size for a good read SNM is an obstacle to robustness and area.

When SRAM reliability and robustness are compared, we simultaneously consider hold, read and write SNM. Usually, since the hold SNM is better than the read SNM, DRV more effectively represents the hold state of the cell. It is, therefore enough to analyze two SNMs: the read SNM and the write SNM. As we observed in Section 2.1.3, the relation between the read SNM and the write SNM has conflicting requirements. In this chapter, we propose a method that resolves the conflicts while considering memory cell power and reliability simulataneously. Specifically, our major contributions are as follows:

- We propose a method that calculates optimal transistor size by considering SNM.

- We improve memory cell reliability and power with optimized pull-up transistor size.

4.1 Motivation

The proper CR and PR affect stability as well as robustness. Many designers choose weak pull-up transistors (PR=0.5) [59, 75] to ensures stable write ability of SRAM. We only consider changing PR since the cell delay and subthreshold leakage are very sensitive to CR. Moreover, we will show that increasing the PR value can lead to enhanced reliability.

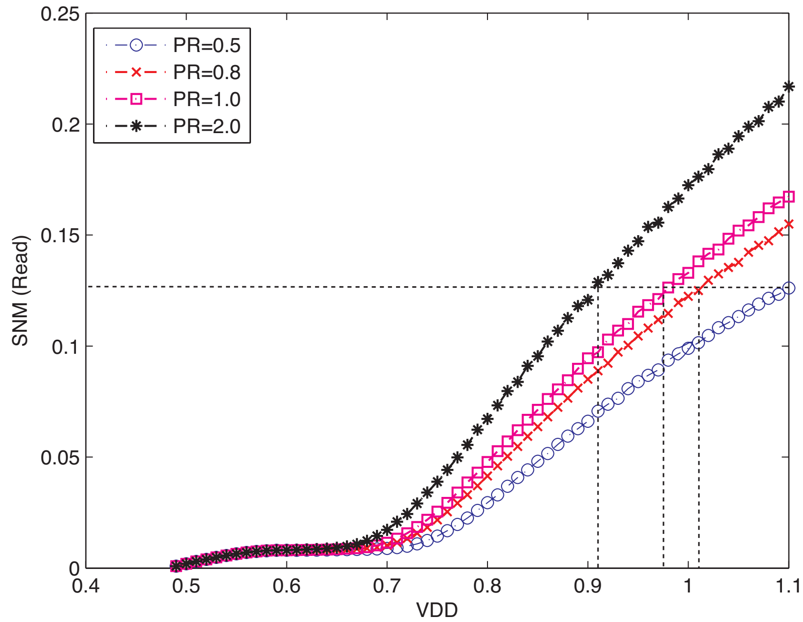


Figure 4.1: Voltage vs. SNM(Read) Using PR=0.5, 0.8, 1.0 and 2.0

Figure 4.1 shows the effect of PR on the SNM at different supply voltage levels. As PR increases, the read stability also increases. According to this relation, it is possible to get the same SNM with a reduced V_{dd} through a very small PR increase. When the voltage

V_{dd} is initially set to a full level ($1.1V$), the SNM value is 0.126. If we increase the PR value from 0.5 to 0.8, the V$_{dd}$ that guarantees the same SNM can be reduced to $1.012V$ allowing the SRAM cells to have lower V$_{dd}$. For example, if we use PR=2.0, the V$_{dd}$ can be lowered from $1.1V$ to near $0.9V$ with the same read SNM as shown in Figure 4.1. However, the cell layout area and the failure rate of the write operation will increase.

4.2 The Upperbound of PR

Enlarging PR within an upperbound is effective for improving the reliability and robustness of the cell operation. As PR increases, the read SNM increases. To analyze the upperbound of PR, the exact maximum PR is defined by the threshold voltage of pull-down transistor V$_{thn}$, the supply voltage V$_{dd}$ and the saturation voltage of pull-up transistor V$_{DSATp}$. Since the output voltage level of the inverter P2-N2 should be less than V$_{thn}$, we can calculate an upperbound for PR. From the equation (2.3), the upperbound of PR is

$$PR \leq \frac{1}{2} \cdot \frac{\mu_n}{\mu_p} \cdot \frac{V_{dd} + 2V_{dd}V_{THn} - 4V_{THn}^2}{K}. \tag{4.1}$$

4.3 SNM-Aware Cell Sizing

Although the upperbound of PR is described, this value is an ideal case for read operations. The large pull-up transistor guarantees the easy readability, but it has a negative effect on the stand-by current and the writeability. We propose to combine the two dependent SNMs in to a single metric. This metric simultaneously minimizes the conflicting

requirements by considering the product SNM

$$\text{SNM}_{product} = \text{SNM}_{read} \cdot \text{SNM}_{write} \tag{4.2}$$

Figure 4.2 shows $\text{SNM}_{product}$ along with its temperature dependence. In general, an in-

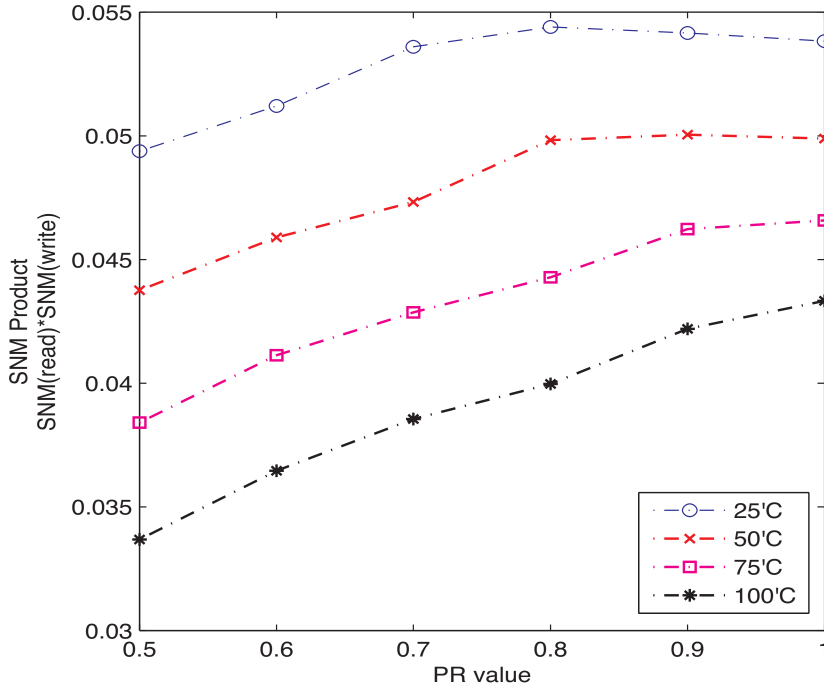


Figure 4.2: Pull-up Ratio (PR) vs. $\text{SNM}_{product}$

creased PR maximizes the product term over the PR range. Additionally, a larger PR reduces the sensitivity of the SNM product to temperature variations. At PR=0.5, the difference between 25°C and 100°C is 32%, but at PR=1.0 the variation in SNM product term is reduced to 19%. Although the product term is maximized at the upperbound at high temperature, the largest PR is not the best choice at low temperature such as 25°C. If we

consider the fact that SRAM circuits spend most of the time in idle mode with a low temperature, it is better to chose based on the common case to reduce area overhead. From this, our method determines a better PR value that guarantees the reliability and robustness.

4.4 Experimental Results

4.4.1 Layout Area Overhead

Since our method employs an up-sized pull-up transistor, the layout area increases. We observe that the overhead of the 6T cell is at most 4.95% as shown in Table 4.1 for various PR ratios. Table 4.2 shows that the area overhead, including peripheral circuitry, is less than 1.95% in practice.

Table 4.1: Cell Layout Area Overhead using PR Value.

PR value	0.5(Base)	0.8	0.9	1.0
Area(nm^2)	0.890	0.910	0.920	0.935
Overhead(%)	0.00%	2.20%	3.30%	4.95%

Table 4.2: Area Overhead of SRAM including Arrays and Peripherals.

SRAM arrays + Peripherals Bits (wordsize)	Area Not-Optimized (nm^2)	Area Our Method (nm^2)	Area Overhead (%)
512 (8)	647.34	659.59	1.89%
512 (16)	627.29	639.54	1.95%
1024 (8)	1213.99	1236.29	1.84%
1024 (16)	1177.52	1199.81	1.89%
1024 (32)	1141.04	1163.33	1.95%
Average (%)	961.44	979.71	**1.91%**

4.4.2 Performance

We observe that the enlarged PR doesn't increase the read delay because the driving strength of the access transistor N3, N4 does not change. However, it can cause the write performance to degrade. This is because as the driving strength of the pull-up transistors P1, P2 becomes larger, I_{DS} of the pull-down transistors N1, N2 increases. As a result, it takes more time to keep the cell value or overwrite the cell. We observe that there is about 3% write delay degradation due to the increased PR and reduced V_{dd} in Table 4.3. However, SRAM performance is typically limited by the read access time, so this will be masked in a typical scenario.

Table 4.3: Performance ($Delay_{write}$) Comparison to Original SRAM.

SRAM Size Bits (wordsize)	Not-Optimized PR:0.5, V_{dd}:1.100V ($Delay_{write}$)	Our Method PR:0.8, V_{dd}:1.012V ($Delay_{write}$)	Speed loss (%)
64 (8)	1.004E-07	1.005E-07	0.10%
256 (8)	3.200E-06	3.301E-06	3.13%
512 (8)	3.301E-06	3.400E-06	3.02%
1024 (8)	6.401E-06	6.601E-06	3.12%

4.4.3 Power Reduction

With a small increase in PR value, we were able to reduce the supply voltage V_{dd} with the same SNM as the original SRAM cell design at full supply voltage. Although the voltage is reduced by 7.96% at 25°C room temperature in the active mode, the SNM is the same as the original as shown in Table 4.4.

In Table 4.4, we also show the total reduction of switching power, $\frac{1}{2}CV^2f$. We observe that our method reduces the active current as much as 7.65% at 25°C. The stand-

Table 4.4: Power Reduction @ 25,50,75 and 100°C using Proposed Methods on a 6T Cell.

Temp (°C)	Method	V_{dd} (V)	SNM (V)	DRV (mV)	Leakage(A) @ DRV	Active(A) @ V_{dd}	Total(A) (I_{leak}+I_{active})
25°C	Not-optimized	1.100	0.126	161.0	1.036E-08	2.842E-07	2.946E-07
	Our method	1.012	0.126	143.4	9.234E-09	2.625E-07	2.717E-07
	Reduction(%)	**7.96%**	**0.00%**	**10.93%**	**10.83%**	**7.65%**	**7.76%**
50°C	Not-optimized	1.100	0.110	173.0	1.659E-08	2.275E-07	2.441E-07
	Our method	0.989	0.110	149.0	1.466E-08	2.154E-07	2.301E-07
	Reduction(%)	**10.09%**	**0.00%**	**13.87%**	**11.63%**	**5.32%**	**5.75%**
75°C	Not-optimized	1.100	0.095	186.0	2.511E-08	1.885E-07	2.136E-07
	Our method	0.985	0.095	160.0	2.257E-08	1.731E-07	1.957E-07
	Reduction(%)	**10.45%**	**0.00%**	**13.98%**	**10.11%**	**8.17%**	**8.40%**
100°C	Not-optimized	1.100	0.082	200.0	3.624E-08	1.665E-07	2.027E-07
	Our method	0.966	0.082	166.0	3.224E-08	1.408E-07	1.730E-07
	Reduction(%)	**12.15%**	**0.00%**	**17.00%**	**11.04%**	**15.45%**	**14.66%**

by leakage power, due to the reduced voltage, is also decreased despite a larger pull-up transistor. We set the supply voltage to the DRV in idle mode which results in an average 10.83% leakage power reduction compared to the original SRAM cell design in Table 4.4.

4.4.4 Gains in Reliability

Our increased PR value reduces the variation in threshold voltage V_{th} while keeping the same static noise margin. The process variation of V_{th} for MOS devices is proportional to $\frac{1}{\sqrt{WL}}$ [60]. We find that the threshold voltage variation in our up-sized cell is less than that of the original cell design. We observe that our method has about 35% reduction in V_{th} variation and an average 7.78% reduction in SNM variation.

Besides the V_{th} variation reduction, we were able to observe that a small increased PR value is helpful to counteract the Negative Bias Temperature Instability (NBTI) [1] which

[1] Generation of interface traps under negative bias condition (V_{gs}=-V_{dd}) shift PMOS transistor V_{th}.

Table 4.5: SNM Variation Comparison between Previous and Our Method.

Type	$\|V_{th}\|$	SNM Not Optimized	$\|V_{th}\|$	SNM Our Method	Improve (%)
Low	0.362	0.320	0.366	0.343	7.49%
Normal	0.384	0.317	0.384	0.341	7.41%
High	0.398	0.315	0.395	0.339	7.31%
Avg.	0.381	0.318	0.381	0.341	7.40%
Var.	3.3E-04	6.4E-06	2.1E-04	5.9E-06	7.78%

increases V_{th} of P type transistor. This degradation reduces the performance of the circuits. To verify this, we estimated the degradation of V_{thp} with a static model [7]. Since our method uses lower V_{dd} and enlarged PR value, it has less V_{thp} degradation than a unoptimized SRAM design over a 10 year lifetime. The V_{th} degradation rate of up-sized pull-up transistor is about 11.3% less than original pull-up transistor as shown in Figure 4.3.

4.5 Conclusions

We presented a method to optimize memory cells considering reliability and power. SNM is an important factor in SRAM design, but most designs follow traditional cell sizing guidelines. However, as technology scales, the traditional cell optimization method does not guarantee the cell's performance, power consumption and reliability. We show an improvement in all three with less than 1.95% area overhead. By increasing the pull-up transistor size, we can reduce the supply voltage as much as 8% with a 10.8% average reduction in leakage power and nearly an 8% reduction in dynamic power. As an additional benefit, we were able to reduce the variation due to V_{th} and SNM variation by 35% and 7.8%, respectively, and significantly increase the lifetime reliability considering NBTI.

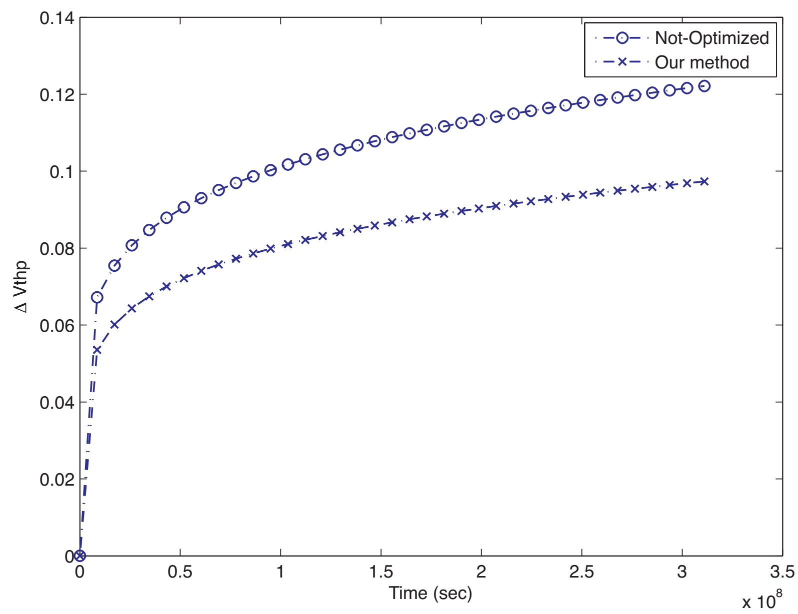


Figure 4.3: Comparison of V_{thp} between Non-optimized and Our method: our method improves NBTI degradation of PMOS transistor compared to non-optimized cell.

Chapter 5

Leakage-Aware Redundancy for Reliable Sub-threshold Memories

Sub-threshold ($V_{dd} < V_{th}$) circuits have been shown to enable substantial reductions in power consumption [13, 66, 81]. The rate of faults, however, increases in sub-threshold circuits due to decreased noise margins and the exponential dependence of current on threshold voltage variation. Previous researchers in sub-threshold circuits have focused on absolute power consumption and have not thoroughly considered the impact of these decisions on circuit yield.

In this chapter, we consider the power consumption of sub-threshold memories while requiring minimum levels of circuit yield. It is easy to show a power reduction while sacrificing yield, but in order to obtain profitability, circuit yield must also be considered. We propose a Monte Carlo framework to analyze the impact of V_{th} variability, memory size, redundancy, and supply voltage on circuit yield. This is enabled with a new fast branch and bound (BB) algorithm to compute the optimal yield obtainable with a repair algorithm.

Our method uses the same circuits for redundancy as have been used with non-parametric memory faults for many decades [16, 72]. We then use this framework to simultaneously analyze the effects of supply voltage and redundancy on the power consumption of sub-threshold memories. This allows the first comparison of sub-threshold memory energy savings with equal yields.

In addition, we propose an enhanced self-repair algorithm that considers leakage reduction in sub-threshold memories by replacing rows and columns that have high leakage and variability. In real memory, leakage power is measured as a total memory leakage using probing pad which is connected to the memory array. To measure row/column leakage separately, we need a practical approach for this. Detailed practical approaches are discussed in Chapter 8.

This leverages a recent development of small programmable fuses that can disable inferior memory cells during manufacturing test [68] to guarantee that the inferior cells no longer contribute to overall array leakage. The inferior cells are replaced as entire rows or columns using the same redundancy circuitry as manufacturing faults.

Specifically, our major contributions are as follows:

- We propose a fast optimal fault repair algorithm based on an efficient branch-and-bound formulation to assess yield.
- We determine the optimal trade-off of supply voltage and redundancy in sub-threshold memories.
- We propose a new method to utilize redundancy for yield and low power simultaneously.

The rest of this chapter proceeds as follows: In Section 5.1, we discuss the problem of sub-threshold reliability. In Section 5.2, we present a fast yield analysis framework. In Section 5.3, we present our method that analyzes the trade-off between V_{dd} and the number of redundancies and enhance this to minimize leakage. In Section 5.4, we propose a method analyzing minimum active voltage and power model using dual V_{dd}. In Section 5.5, we present our experimental results and analysis. Finally, Section 5.6 concludes the chapter.

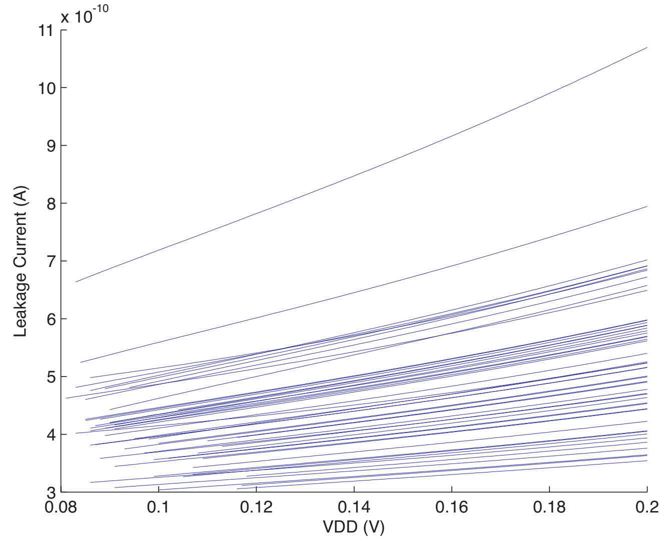
5.1 Sub-threshold Reliability

Sub-threshold leakage current can be modeled as

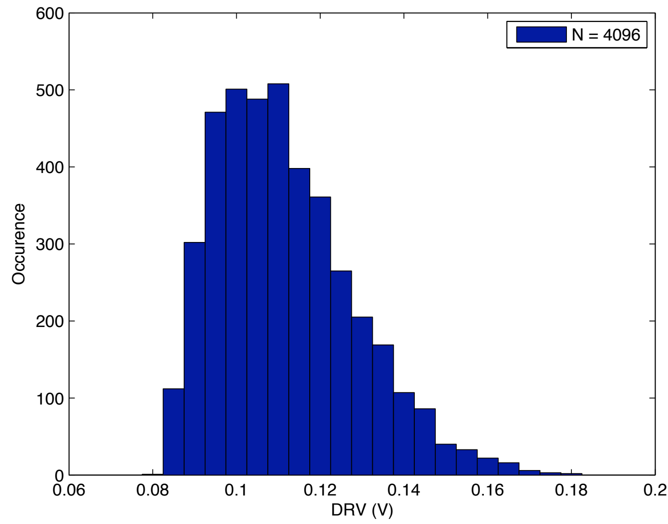
$$I_{leakage} = \left(\frac{W}{L}\right) I_s \left\{1 - e^{\frac{-V_{dd}}{V_T}}\right\} e^{\frac{-\left(V_{th}+V_{off}\right)}{nV_T}} \tag{5.1}$$

where W is transistor width; L is transistor length; V_T is the thermal voltage; V_{th} is the threshold voltage; and V_{off}, I_s and n are empirical constants. It is important to note that V_{dd} is an exponential term, so that lowering V_{dd} can result in exponential leakage reduction in the super-threshold region. In the sub-threshold region, however, the leakage current is roughly linear with the supply voltage V_{dd} because e^{-V_{dd}/V_T} is nearly zero and the offset voltage V_{off} is small [10].

Process variation of V_{th} due to random dopant fluctuation [60], however, becomes significantly more problematic in the sub-threshold region. While the effect of V_{dd} scaling on leakage is nearly linear in the sub-threshold region, the impact of V_{th} variation is not. Figure 5.1(a) shows how sub-threshold supply scaling effects cell leakage for a number of memory cells with variability. Each cell has a different sensitivity with respect to V_{dd} in



(a) V_{dd} vs. Leakage current



(b) DRV distribution

Figure 5.1: Sub-threshold V_{dd} effect on 1K cell's leakage and DRV distribution of 4K cell with PTM [4] model

the sub-threshold region due to V_{th} mismatch in the memory cell. More importantly, the leakage can vary over a large range at a given supply voltage.

This variation becomes even more significant when noise margins are considered. The Data Retention Voltage (DRV) is the minimum supply voltage capable of holding the correct value of SRAM in stand-by. Higher supply voltages are required for read and write operations, but this work focuses on stand-by optimization since it is the majority of power in many applications. If the supply voltage is less than a cell's DRV ($V_{dd} < DRV$), the cell will fail to store the correct value in stand-by mode. DRV variation is depicted in Figure 5.1(b) for 4K memory cells. The long tail of the distribution means that very few cells will have a high DRV and limit the overall supply voltage. Reducing V_{dd} introduces exponentially more faults due to the log-normal-like distribution of the DRV.

SRAM designers typically select the supply voltage as the maximum DRV along with a guard band voltage to guarantee reliability and reasonable yield [62]. However, this maximum DRV is determined by very few cells and is therefore suboptimal for overall array leakage. Instead, we propose that redundant rows and columns can be used to replace these parametric cell failures and enable a reduction of the maximum DRV for an entire array. These redundant rows/columns require extra power and area, but will enable power reduction of the entire array through supply voltage reduction. The question remains as to how much redundancy is optimal when considering overall power and reliability simultaneously.

5.2 Yield Analysis Methodology

This section discusses the framework to analyze the trade-off between redundancy and power while considering DRV variation, redundancy, fault-repair and yield. The basis of our method is a Monte Carlo (MC) simulation and optimal fault repair of multiple memories at a range of supply voltages to determine manufacturing yield. Since MC of memories is challenging due to large run-times, this section focuses on speeding up the bottlenecks.

5.2.1 Yield Simulation

An overview of our MC framework is shown in Figure 5.2. A DRV_{map} is defined as a $N \times M$ array of memory cells selected from a pre-computed pool. Multiple maps are created using sampling with replacement from the pool. An index (i, j) of DRV_{ij} denotes the corresponding cell in a DRV_{map} such that $i \in [0, N-1], j \in [0, M-1]$.

We use MC simulation considering the V_{th} variation of a memory cell to pre-compute the DRV pool through simulation. This pool has independent transistor variations and characterizes the *DRV* of each cell, cell leakage at each DRV, and the cell leakage at a given supply voltage V_{dd} as a 3-tuple:

$$(DRV_{ij}, I_{ij}(DRV_{ij}), I_{ij}(V_{dd})).$$

The DRV pool will have a distribution similar to Figure 5.1(b).

Each DRV pool entry has two leakage terms $I_{ij}(DRV_{ij})$ and $I_{ij}(V_{dd})$. These two terms indicate the cell leakage at DRV and V_{dd}, and enable us to estimate the cell leakage

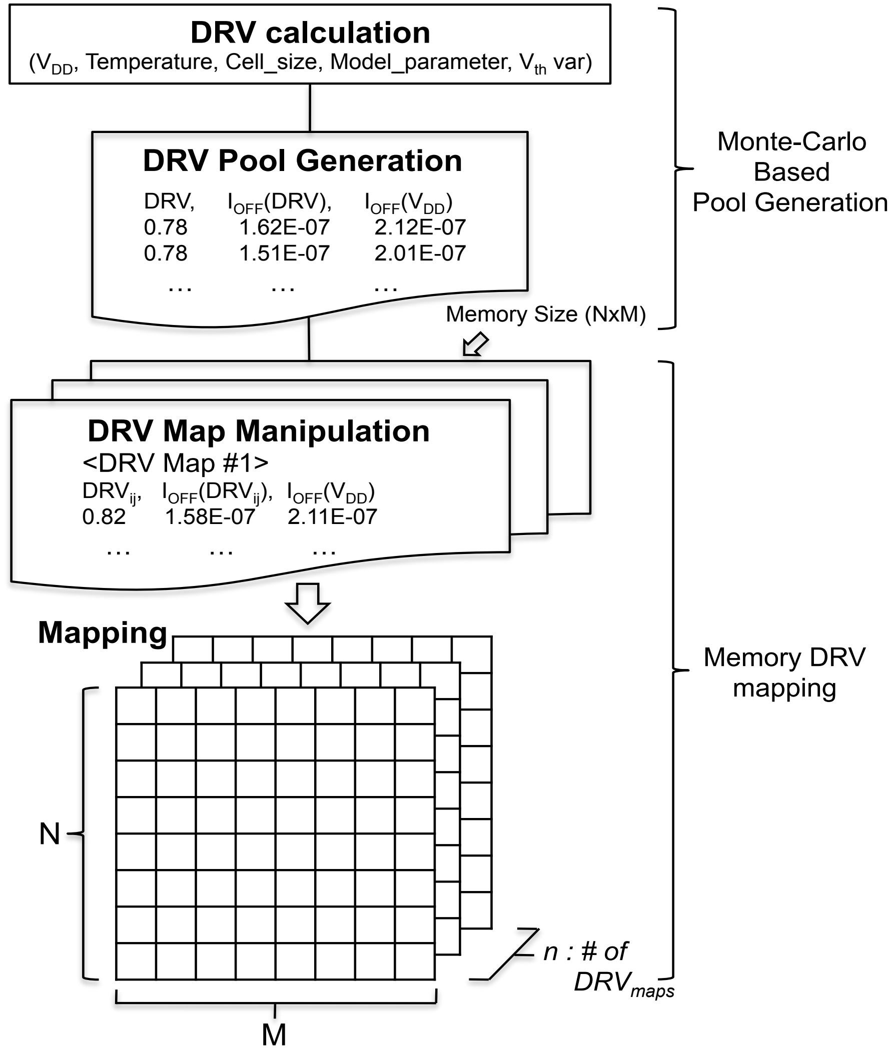


Figure 5.2: Monte Carlo Framework for DRV_{map} manipulation

quickly. They will be further discussed in Section 5.2.3.

To compute yield, we generate n DRV_{map} arrays and perform self-repair given a fixed supply voltage V_{dd} and a fixed number of row (nr) and column (nc) redundancies. The fast optimal self-repair algorithm is discussed in Section 5.2.2. The yield is the fraction of repairable arrays out of the n generated.

5.2.2 Optimal Fault Repair Analysis

We use an optimal fault repair algorithm in order to accurately estimate the optimal yield during manufacturing. While actual self-repair algorithms may not be optimal, our approach gives the best-case yield in order to quantify this effect simultaneously with the power reductions.

In sub-threshold circuits, the DRV voltage has a large tail and when DRV is pushed very low, a large number of faults occur. The previous methods for repair analysis [44] focused on fault repair with a limited number of redundancies and low fault densities. These prior works introduced the *must-repair* criteria to reduce the problem size, formulated the problem as search tree, and exhaustively searched for a feasible solution of the remaining faults. This can potentially not find a feasible solution and can require large run-times with highly dense faults.

We propose a branch-and-bound method that scales well even with highly dense faults and large levels of redundancy. Our method relies on a depth-first search and we propose a new pruning criteria that efficiently bounds the infeasible solution spaces. We also present heuristics that guide the solution towards feasible solutions more quickly.

Specifically, our method's pruning operation considers the remaining unused re-

dundancies and the remaining fault pattern to eliminate portions of the search space that are guaranteed to have no solution. From [58], a *must-repair* is defined as

Observation 1 *For a memory with nr redundant rows and nc redundant columns the following repair decisions are mandatory: If there are more than nc faults in a row, then there are not enough columns to cover all the faults, and a row must be selected for repair. Similarly, more than nr faults in a column require a column repair.*

We make a second observation that extends this idea to a *cannot-repair* situation:

Observation 2 *For a memory with nr redundant rows and nc redundant columns, the total number of available redundancies must be larger than the number of separately located faults for a feasible solution to exist.*

A fault is separately located if it is not in the same row or column as another fault. Based on the contrapositive of the proposition in Observation 2, we derive our pruning criteria:

Criterion 1 *Let the available number of redundancies be (nr, nc) and define $N_{separate}$ to be the number of separately located faults. A feasible solution can not exist if and only if $|nr| + |nc| < N_{separate}$.*

The proof of above criterion is desrcibed both way ($\Leftrightarrow$) as follow. ($\Rightarrow$) trivial. ($\Leftarrow$) For proof, take a contraposition of original criterion. If the number of redundancies ($|nr| + |nc|$) is more than the sum of number of rows and columns ($N + M$), then feasible solution should exist because of plenty of redundancies ($|nr| + |nc| \geq M + N$). This satisfies $|nr| + |nc| \geq M + N \geq min(M, N)$. Also $N_{separate}$ is always less than $min(M, N)$

by definition of $N_{separate}$, it results in $|nr| + |nc| \geq N_{separate}$ eventually. Therefore, the above criterion places a conservative bound on the number of separately distributed faults that are feasibly repaired.

Another key component of our optimal repair algorithm is the directed search. We utilize the insight from previous greedy algorithms that it is often advantageous to repair more faults quickly to reduce the problem size. Therefore, we use the *repair most* concept to determine which fault to repair next and whether to use a column or a row to repair that fault. The *RowDegree* (*ColDegree*) of a fault is the number of faults in the same row (column). We use the *RowDegree* and *ColDegree* of each fault to direct the search. The maximum degree determines which fault to repair first and the row or column degree determines how to first try to repair the fault. This heuristic does not affect the optimality because if a feasible solution is not found, the search algorithm will back-track to find a feasible solution.

Algorithm 1 summarizes our recursive algorithm called *FastRepair*. As an input, the algorithm takes a number of redundant rows (nr) and columns (nc), a DRV_{map} with process variations, and the desired V_{dd}. The output of the algorithm is whether the given memory can be repaired using the provided redundancies. The algorithm starts by calculating the number of faults using *ScanFaults* and *SizeOf* with given DRV_{map} and V_{dd} in Lines 1-2. In Lines 3-6, we perform the trivial base case feasibility check to see if there are remaining faults and no unused redundancies. In Lines 7-10, we use Criterion 1 as an additional pruning step. If Criterion 1 holds ($N_{faults} > |nr| + |nc|$), it is not possible to find a feasible solution and we return $Fail$. For simplicity, we only check Criterion 1 when the maximum degree of all remaining faults is one ($MaxDegree(Faults_{list}) = 1$), but it also holds when we count the total separately located faults in any fault map.

Algorithm 1 $FastRepair(nr, nc, DRV_{map}, V_{dd})$

Require: Available $|nr| + |rc|$; DRV_{map}; V_{dd}
Ensure: Determine feasibility of repairing faults

```
1:  Faults_list ⇐ ScanFaults(DRV_map, V_dd)
2:  N_faults ⇐ SizeOf(Faults_list)
3:  // Prune trivial base case
4:  if (N_faults > 0)&(nr = 0)&(nc = 0) then
5:      Return Fail
6:  end if
7:  // Prune based on criteria 1
8:  if MaxDegree(Faults_list)=1 & N_faults > |nr| + |nc| then
9:      Return Fail
10: end if
11: // Recurse using "repair most" criteria
12: Faults_list ⇐ SortByMaxDegree(Faults_list)
13: for n_{i,j} ∈ Faults_list do
14:     if RowDegree(n_{i,j}) > ColDegree(n_{i,j}) then
15:         Replace row i and update DRV_map
16:         if FastRepair(nr − 1, nc, DRV_map, V_dd) then
17:             Return Success
18:         end if
19:         Restore row i and update DRV_map
20:         Replace col j and update DRV_map
21:         if FastRepair(nr, nc − 1, DRV_map, V_dd) then
22:             Return Success
23:         end if
24:         Restore col j and update DRV_map
25:     else
26:         // Replace column then row similar to lines 14-24.
27:     end if
28: end for
29: if N_faults = 0 then
30:     Return Success
31: else
32:     Return Fail
33: end if
```

Lines 11-28 show the core of the recursive algorithm. The fault locations are first sorted in-place according to the decreasing maximum of the $RowDegree$ and $ColDegree$ in the procedure *SortByMaxDegree*. This determines the decreasing degree order in which we try to repair the faults. For each fault $n_{i,j}$ in the sorted list, we determine whether *RowDegree* or *ColDegree* is more significant and repair the appropriate row or column first. The algorithm replaces the faulty row/column and recursively calls FastRepair with updated nr, nc values and a DRV_{map} with the redundant row/column substituted in place of the faulty row/column. If the recursive call obtains $Success$, we can return the solution immediately. Otherwise, we must undo the replaced row/column and recurse while replacing with a column/row instead. The preference of fault repair ordering and row/column selection is the essence of the *repair most* guidance. If no solution can be found by replacing all of the faults, then the algorithm was not successful.

Our method is optimal because it searches all possible replacements in the worst case. In doing this, any feasible solution will be found if it exists. This is possible because the problem size is reduced by pruning infeasible solution spaces with Criteria 1. Therefore, Algorithm 1 can provide the optimal fault repair with less time on average than prior methods.

5.2.3 Leakage Calculation

Given a supply voltage V_{dd} for entire array, we utilize a fast leakage estimate. Each cell's leakage under various supply voltages in the sub-threshold region is approximately linear as shown in Figure 5.1(a). Given this, we calculate the leakage $I_{ij}(x)$ of each cell at supply voltage x using $I_{ij}(DRV_{ij})$, $I_{ij}(V_{dd})$ and

α=$\frac{I_{ij}(V_{dd})-I_{ij}(DRV_{ij})}{V_{dd}-DRV_{ij}}$ according to

$$I_{ij}(x) = \alpha(x - DRV_{ij}) + I_{ij}(DRV_{ij}) \tag{5.2}$$

The $I_{ij}(DRV_{ij})$ and $I_{ij}(V_{dd})$ values are pre-computed for each DRV pool sample as discussed in Section 5.2.1. The cell leakage $I_{ij}(x)$ at any supply voltage x can be quickly calculated using each cell's DRV_{ij}. Figure 5.3 shows the cell leakage of 16K

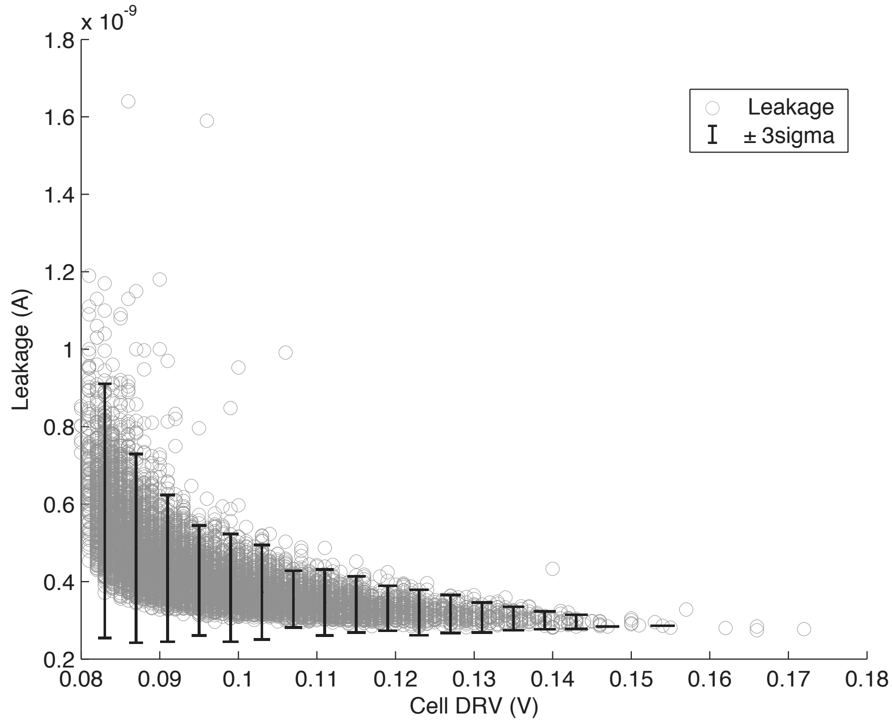


Figure 5.3: Cell DRV vs. Cell leakage at V_{dd}=130mV on 16K sample cells

cells at 130mV. Although the DRV of some cells is above 130mV, these will be repaired with redundant cells. The other cells operate reliably because of the supply voltage $V_{dd} > DRV$. Since each cell is individually characterized in the DRV pool, the effect of variation

on leakage is accounted for in our models. The total memory leakage of a DRV_{map} at a given $V_{dd} = x$ is computed by summing each cell's leakage $I_{ij}(x)$ as

$$I_{mem_k}(x) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} I_{ij}(x) \tag{5.3}$$

where $k \in [0, n-1]$ is one of the MC DRV_{map}. The replaced rows and columns are excluded since they will be removed using fuses during manufacturing test [68]. Since the yield calculation procedure is performed on a number n of DRV maps, the expected value of leakage at supply voltage x is computed as

$$E(I_{mem})_x = \left(\frac{1}{n}\right) \sum_{k=0}^{n-1} I_{mem_k}(x) \tag{5.4}$$

This will be used in the next section for leakage-aware redundancy usage.

5.3 Leakage-Aware Redundancies

Using the previous yield simulation, self-repair analysis and leakage estimation framework, we can now consider the impact of redundancy on yield and leakage. Previous works did not analyze the power consumption after the self-repair process because of the problematic time complexity of self-repair. Nor did previous works consider the impact of stand-by power scaling on yield. In this section, we derive the minimum V_{dd} to attain a required yield in Section 5.3.1 and then reduce the memory leakage without increasing yield loss by considering the redundancy level. Furthermore, we propose an additional usage of redundancy to improve both yield and the leakage in Section 5.3.2.

5.3.1 Supply Voltage Lower Bound for Required Yield

Using our $FastRepair$ algorithm, we can study the effect of V_{dd} and redundancy on yield. Figure 5.5 shows the yield curves for a 1K memory over a range of stand-by V_{dd} levels with differing amounts of row and column redundancy (nr, nc). From the figure, it is clear that no redundancy (0,0) has low yield at low voltages whereas the most redundancy (5,5) achieves 99.9% yield around $137mV$. In Figure 5.5, we also observe that the required V_{dd} to obtain a yield, V_{stdby}, increases as the number of redundancies are reduced. For example, V_{stdby} for 99.9% yield is near $141mV$ with $(4, 4)$, but increases to $151mV$ with $(2, 2)$ redundancies. If we provide a V_{dd} larger than V_{stdby}, we can easily achieve the required yield, but this will waste power.

To directly calculate V_{stdby}, we did basic nonlinear fitting using V_{dd} and the available number of redundancies. Figure 5.4 shows the accuracy of our yield model to yield simulation result for V_{stdby} calculation. We observe that each simulation yield data on various available number of redundancies (2,2)-(8,8) is fitted well with the analytic model described as below.

Given the previous data, the yield at the supply voltage x can be expressed as

$$Yield(x) = \kappa_1 \cdot exp(\kappa_2 \cdot x) + \kappa_3 \tag{5.5}$$

where $\kappa_1, \kappa_2, \kappa_3$ are nonlinear fitting parameters. If we assume that memory design requires 99.9% yield (much higher may be used in practice), we can calculate V_{stdby} by solving

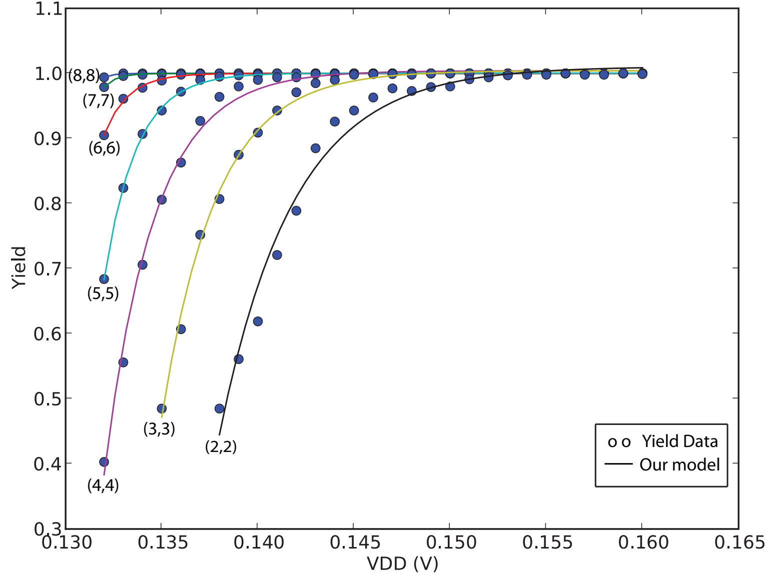


Figure 5.4: Our yield model matches simulation data for V_{stdby} calculation with good accuracy.

$Yield(x) \geq 0.999$ and obtaining

$$V_{stdby} \geq \frac{1}{\kappa_2} \cdot log(\frac{0.999 - \kappa_3}{\kappa_1}) \tag{5.6}$$

The V_{stdby} provides a reliable supply voltage satisfying a yield requirement and enables power trade-off comparisons. More results will be discussed in Section 5.5.

5.3.2 Leakage Optimization

While the previous analysis showed that increased redundancy can enable reduced supply voltage and therefore decreased leakage, we now propose another strategy for leakage optimization. First, we observe in Figure 5.3 that not only does the leakage in-

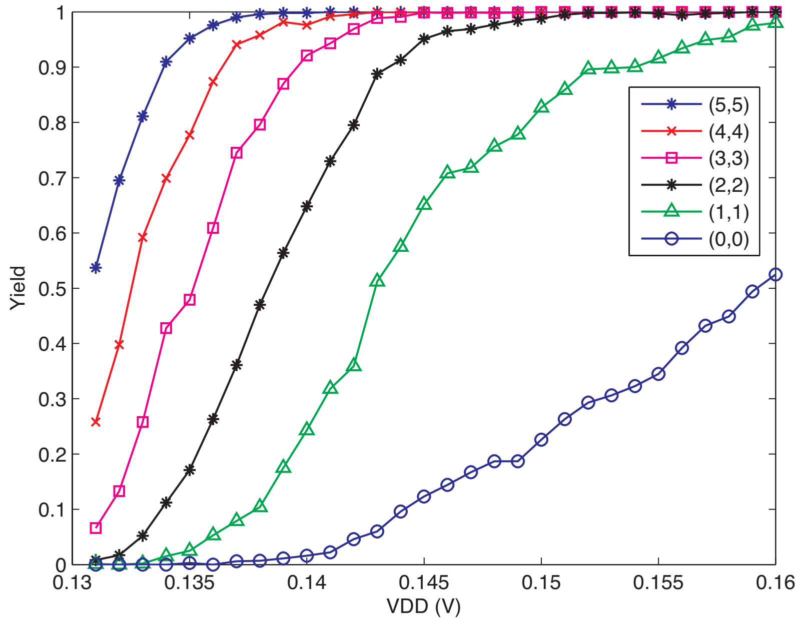


Figure 5.5: V_{dd}, redundancies effect on yield with 1K SRAM

crease on average with very low DRV values, but the variation in the leakage also increases. From this, we can conclude that lowering the maximum DRV of memory cells is good for lowering the voltage of an entire array, but if the DRV of a cell is too low, it may actually increase the expected leakage of an array.

The redundancy in Section 5.3.1 proposed to utilize redundancy to lower the supply voltage only. However, we now propose that we can replace cells with very low DRV to reduce the leakage and leakage variability of a memory array. Since the low DRV cells are leaky, using redundancies on overly leaky cells can efficiently reduce the leakage. If we increase V_{dd} by a small amount, the number of faults will reduce and we can use our redundancies to replace the leaky cells instead of faulty ones. The inferior leaky cells are replaced as entire rows or columns using the same redundancy circuitry as manufacturing

faults.

We statically allocate a portion of the redundancies for leakage reduction and the remaining portion is for fault repair and supply voltage reduction. Since the number of redundant rows/columns is quite low, we can perform a simple enumeration of the possible allocations and select the best combination. Each allocation results in a different V_{stdby} because it is calculated with the redundancies allocated for supply reduction as done in Section 5.3.1. To reduce leakage, we greedily select the leakiest cell according to $I_{ij}(V_{stdby})$ and replace it with a row or column that provides the maximal reduction in overall leakage, $I_{mem}(V_{stdby})$. It is not possible to measure row leakage and column leakage separately using previous approach that uses one probing pad per a memory array. We assume that additional power lines (V_{dd} and GND) are railed to each row and column for measuring row and column leakage. The replacement process is finished when there are no more available redundancies for leakage reduction.

As a specific example, if we are given a memory with (6,6) redundancies, we can use (4,4) of the redundancies for fault repair and (2,2) of the redundancies for leakage reduction or, instead, we can use (3,3) for fault repair and (3,3) for leakage. Each case uses all (6,6) redundancies. Our method will select the allocation with the better leakage.

5.4 Supply Voltage for Active Mode Power

In Section 5.3, we analyzed the optimal voltage for a memory in stand-by operation. This section considers the read/write operation in addition to the stand-by operation. This section describes how to calculate the minimum operating voltage and total power

while simultaneously considering leakage and dynamic power.

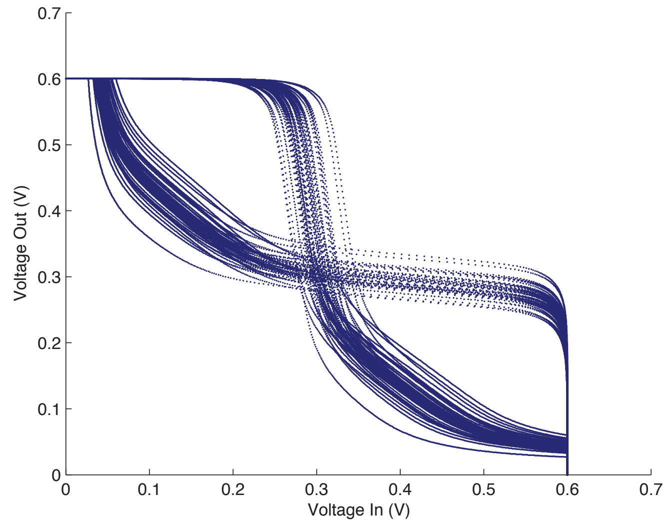
5.4.1 Minimum Read/Write Voltage in Sub-threshold Regions

We first observe that a memory's minimum stand-by required voltage V_{stdby} can be calculated as in Section 5.3 and then extend the approach to determine the minimum active operating voltage V_{act}.

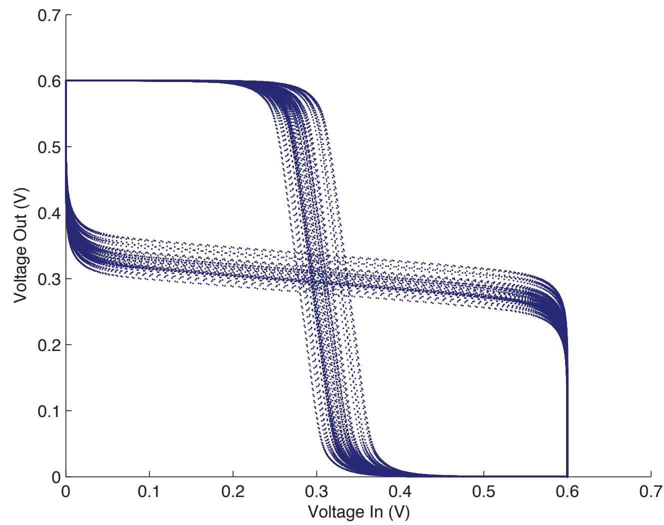
To determine the minimum operating voltage, we must analyze the 'butterfly curve' which determines the SNM of each cell. Figure 5.6(a) and Figure 5.6(b) show the read SNM of both 6T and 8T memory cells with variation. We observe that the 8T cell's SNM is larger than the 6T cell's SNM at the same voltage level. This means that a smaller voltage is necessary for a same read operation capability in 8T cells without causing a destructive read compared to 6T cell. It also means that the 8T cell's read operation has more potential for power reduction in active mode despite both the 6T and 8T cell having a similar DRV distribution.

Our method calculates each cell's minimum operating voltage while we analyze the minimum DRV voltage in Section 5.2. We sweep the supply voltage and analyze the SNM at each supply voltage. The minimum operating voltage for a cell is the voltage that causes the read SNM to be near zero. Figure 5.7(a) and Figure 5.7(b) show the minimum operation V_{dd} distribution of 6T and 8T cells of a sample pool with V_{th} variation. It is proven that read SNM of 6T cell is normal distribution [18]. The minimum operating voltage V_{act} for the memory array is the maximal voltage in the distribution's tail.

We observe that the maximum necessary voltage for 8T cells is $440mV$ in Figure 5.7(b), but this voltage for 6T cells is about $650mV$ in Figure 5.7(a). This means that

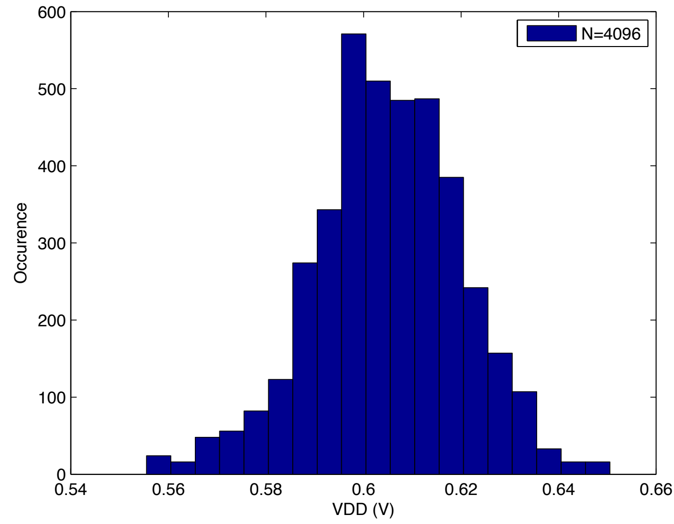


(a) 6T cell's read SNM

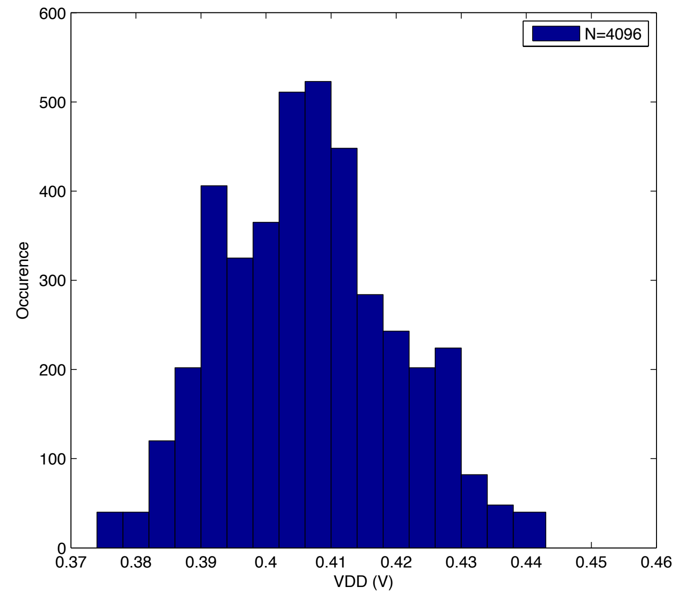


(b) 8T cell's read SNM

Figure 5.6: 8T cells need a smaller voltage to have same read stability according to SNM.



(a) 6T cell's minimum operating V_{dd} distribution.



(b) 8T cell's minimum operating V_{dd} distribution.

Figure 5.7: Active operating voltage V_{act} distribution of 8T cells are significantly lower than 6T cells.

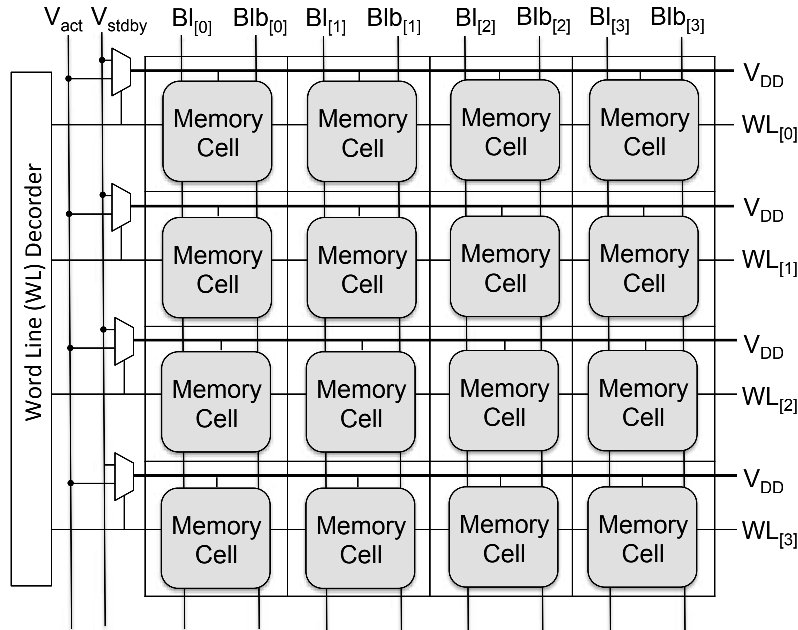


Figure 5.8: Memory architecture using the optimal voltage V_{stdby} and V_{act} for idle mode and active mode

8T cells have an advantage over 6T cells in terms of active power reduction because of separated read line. Also, we observe that the 6T cell's minimum operation voltage distribution has a larger V_{act} variation than 8T cell's V_{act}. For example, 6T cell's V_{act} difference is $100mV$, but 8T cell's V_{act} difference is $70mV$.

5.4.2 Power Model using V_{stdby} and V_{act}

The total power consumption is calculated based on the assumption that memory uses dual voltages (V_{stdby} and V_{act}) for reliability and power reduction according to its operating mode. This is a common practice [21,40] and is illustrated in Figure 5.8 for a memory array using the two supply voltages. The operating voltage (V_{stdby} or V_{act}) is selected for each row using the word-line decoder. The correct usage of V_{stdby} and V_{act} guarantees the

reliability in both standby mode and low power in active mode simultaneously.

Using V_{stdby} for stand-by operation and V_{act} for read/write operation, the total memory power is

$$P_{mem}(V_{stdby}, V_{act}) = P_{leak}(V_{stdby}) + P_{dyn}(V_{act}) \tag{5.7}$$

where $P_{leak}(V_{stdby})$ and $P_{dyn}(V_{act})$ are the leakage power and the dynamic power of the memory array using the two voltages (V_{stdby}, V_{act}) according to the operating mode. Given a $N \times M$ memory array, $P_{leak}(V_{stdby})$ is calculated from Equation (5.4) as

$$P_{leak}(V_{stdby}) = V_{stdby} \sum_{j=0}^{M-1} \left(\sum_{i=0}^{k-1} I_{ij}(V_{stdby}) + \sum_{i=k+1}^{N-1} I_{ij}(V_{stdby}) \right) \tag{5.8}$$

where $I_{ij}(V_{stdby})$ is the each cell's leakage current and is extended with $N-1$ rows and M columns to calculate the memory leakage current except for the accessed row $k \in [0, N-1]$ that consumes dynamic power. The dynamic power $P_{dyn}(V_{act})$ is calculated using the dynamic power $\frac{1}{2}CV_{act}^2 f$ with each cell's capacitance C_j and operating frequency f as

$$P_{dyn}(V_{act}) = \sum_{j=0}^{M-1} \frac{1}{2} C_j V_{act}^2 f. \tag{5.9}$$

Combining Equation (5.8) and Equation (5.9), we can substitute $P_{leak}(V_{stdby})$ and $P_{dyn}(V_{act})$

of Equation (5.7).

$$P_{mem}(V_{stdby}, V_{act}) = V_{stdby} \sum_{j=0}^{M-1} \left(\sum_{i=0}^{k-1} I_{ij}(V_{stdby}) + \sum_{i=k+1}^{N-1} I_{ij}(V_{stdby}) \right) + \sum_{j=0}^{M-1} \frac{1}{2} C_j V_{act}^2 f. \qquad (5.10)$$

Equation (5.10) implies that the total memory power is the output of various parameters such as V_{stdby}, V_{act} ($V_{act} > V_{stdby}$) and f. The optimal voltage V_{stdby} and V_{act} along with the frequency f effect will be further discussed with the experimental result in Section 5.5.

5.5 Experimental Results

5.5.1 Experimental Setup

Our method generates memories with random variation using a 16K SRAM cell pool. We analyze various SRAM sizes from 1K to 16K. We focus on small banks of memories that may be combined into larger memories. The simulations are based on the 45nm PTM technology models with a temperature of 25°C [4]. We consider memory array without ECC (Error Correction Code), and we assume that transistors have independent $\pm 15\%/3\sigma$ variation of the nominal V_{th}. The pull-up/pull-down SRAM transistor width size ratio is 0.5 and $\frac{PR}{CR}$=$\frac{90nm/45nm}{180nm/45nm}$ with the same length [59]. We perform Monte Carlo simulation and repair on *n=1,000* different DRV maps using *FastRepair*. The simulation framework is written in Python and executed on an Ubuntu 10.04 system with 2.2GHz CPU and 4GB memory.

5.5.2 Yield Accuracy and CPU time

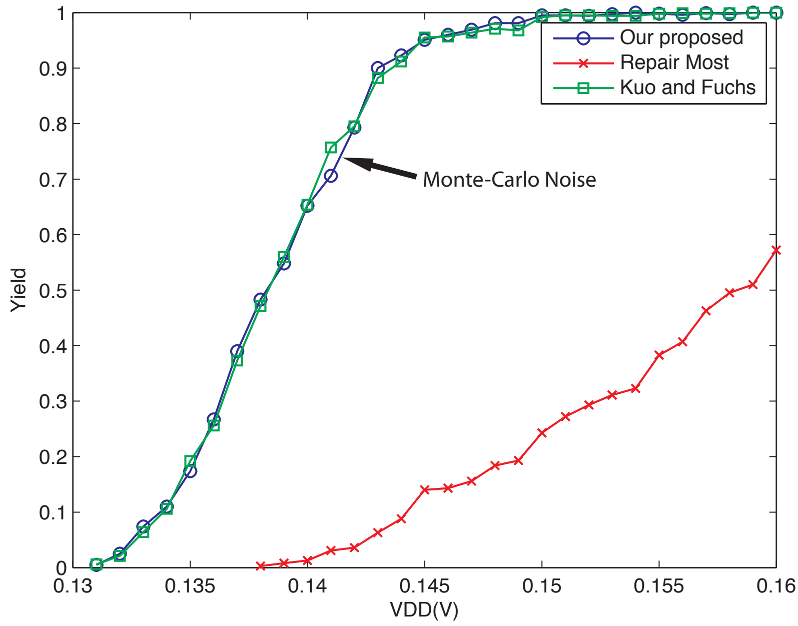
Figure 5.9(a) shows the yield comparison of our method with the greedy Repair Most (RM) method and Kuo and Fuchs' exact method [44] when (2,2) redundancies are available. Our results are very similar to the exhaustive search by Kuo and Fuch and significantly more accurate than the RM greedy approach. There is a slight difference between our method and Kuo and Fuchs near $142mV$, but this is noise due to MC.

Figure 5.9(b) shows the CPU time on average of each repair algorithm with $(6, 6)$ redundancies over the different voltages. We observe that our method is much faster than Kuo and Fuchs' method. The percentage of faulty cells increases exponentially for very low supply voltages. Both RM and Kuo and Fuchs's method take more time than our method with highly dense faults since ours quickly prunes infeasible solutions. RM, however, is faster for low density of faults while our algorithm is similar to Kuo and Fuchs' method.

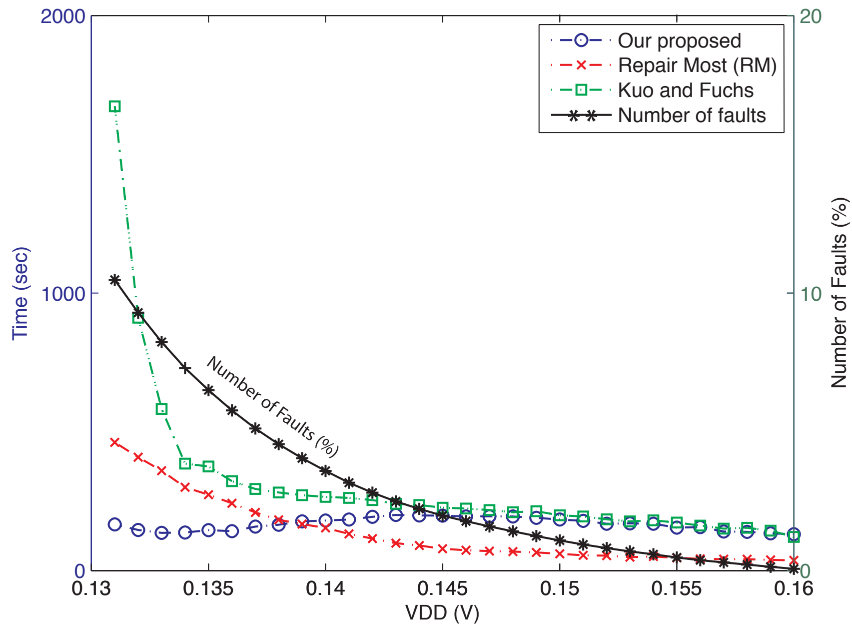
5.5.3 Supply Voltage and Redundancy

Figure 5.10 shows the leakage at several required yields for a 1K SRAM using only supply voltage reduction and fault repair. Each contour represents a different required yield 50-99.9%. Increasing the level of redundancy significantly reduces the leakage. With (5,5) redundancies, a 10% leakage reduction is expected with 99.9% yield when compared to the leakage of a solution without redundancy.

Moreover, we observe that the effect of redundancy on yield is diminishing with highly redundant arrays. The yield difference between 99.9% and 70% with (15,15) redundancies is significantly smaller than the same yield difference with (0,0) redundancies.



(a) Yield analysis comparison



(b) CPU time comparison

Figure 5.9: Comparison with previous methods with 1K SRAM

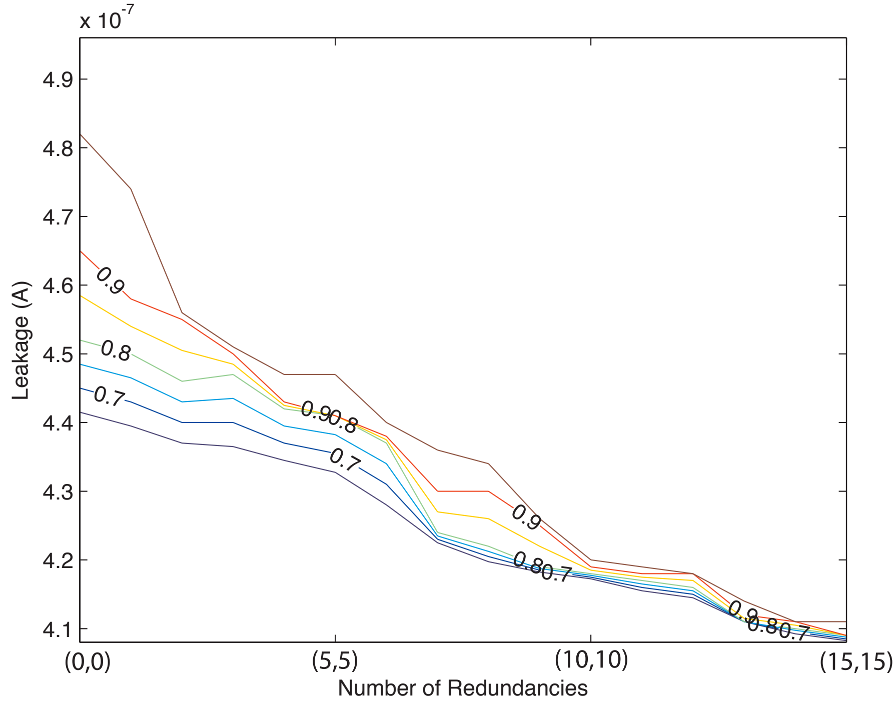


Figure 5.10: Redundancy Effect on Leakage at Required Yields for 1K SRAM

5.5.4 Leakage Reduction

Table 5.1 shows the stand-by leakage and corresponding V_{stdby} for 99.9% yield on a variety of SRAM sizes. We compared two optimization strategies: 1) our proposed method with redundancy and supply voltage reduction only (Section 5.3.1) and 2) our proposed method with both supply voltage and low-DRV replacement (Section 5.3.2). The baseline is an SRAM without any redundancies and worst-case supply voltage to attain the required yield.

According to Table 5.1, supply voltage optimization with redundancies can reduce the stand-by leakage by an average of 10.92% when there are (6,6) redundancies available and 99.9% yield is required. Moreover, we are also able to reduce the stand-by leakage by average 13.72% if we replace low DRV cells instead of just faulty cells to lower

the supply voltage. This additional leakage optimization method is more efficient in small memory sizes (1-2K) since there are fewer cells and likely to be fewer high DRV cells. It is therefore more advantageous to use all of the redundancies (6,6) for leakage reduction rather than supply voltage reduction. We also observe that the lowest V_{stdby} for 99.9% yield is not always best to reduce the stand-by leakage in these small memories. Table 5.1 shows that V_{stdby}=162mV for a DRV-optimized 1K SRAM is larger than the V_{stdby}=135mV for a supply voltage-only optimization. However, the stand-by leakage is improved by almost two times. This shows that the lowest V_{dd} is not always the best when considering fault-repair and overall yield. Table 5.2 shows the experimental result when 8T cell is applied instead of 6T cell. In memory array using 8T cell, we observed similar results with the memory array using 6T cell. V_{DD} optimization only and both (V_{DD} + DRV) optimization of Table 5.2 reduce the leakage by average 9.53% and 13.44% respectively.

The 4K memory shows the cross-over point where it is best to use most redundancies (5,5) for supply voltage reduction and the remainder (1,1) for low DRV replacement in 6T cell. Also, 4K memory also shows the cross-over point in 8T cell. The half of redundancies (3,3) for supply voltage reduction and the rest of redundancies (3,3) for low DRV replacement give the minimum leakage. The large size memories (8-16K) best utilize the available redundancies by lowering the overall array supply voltage in both cases.

Table 5.1: Comparison of Leakage at 99.9% Yield when (6,6) redundancies are available. (6T cell)

	SRAM w/o redundancies		Supply Voltage Optimization				Supply and Low DRV Optimization				
Size	V_{stdby} (V)	Leakage (A)	V_{stdby} (V)	Leakage (A)	For Faults	Improvement(%)	V_{stdby} (V)	Leakage (A)	For Faults	For Leakage	Improvement(%)
1K	0.183	5.01E-07	0.135	4.56E-07	(6,6)	8.95%	0.162	4.27E-07	(0,0)	(6,6)	15.92%
2K	0.183	1.03E-07	0.135	9.08E-07	(6,6)	11.59%	0.162	8.68E-07	(0,0)	(6,6)	15.49%
4K	0.183	2.05E-06	0.135	1.81E-06	(6,6)	11.77%	0.138	1.74E-06	(5,5)	(1,1)	14.91%
8K	0.183	4.05E-06	0.135	3.60E-06	(6,6)	11.03%	0.135	3.60E-06	(6,6)	(0,0)	11.03%
16K	0.183	8.05E-06	0.135	7.14E-06	(6,6)	11.27%	0.135	7.14E-06	(6,6)	(0,0)	11.27%
			Average Improvement(%)			10.92%	Average Improvement(%)				13.72%

Table 5.2: Comparison of Leakage at 99.9% yield when (6,6) redundancies are available. (8T cell)

	SRAM w/o redundancies		Supply Voltage Optimization				Supply and Low DRV Optimization				
Size	V_{stdby} (V)	Leakage (A)	V_{stdby} (V)	Leakage (A)	For Faults	Improvement(%)	V_{stdby} (V)	Leakage (A)	For Faults	For Leakage	Improvement(%)
1K	0.183	4.82E-07	0.133	4.38E-07	(6,6)	8.99%	0.181	3.83E-07	(0,0)	(6,6)	20.46%
2K	0.183	9.58E-07	0.133	8.68E-07	(6,6)	9.42%	0.181	8.05E-07	(0,0)	(6,6)	16.04%
4K	0.183	1.91E-06	0.133	1.73E-06	(6,6)	9.63%	0.145	1.70E-06	(3,3)	(3,3)	10.89%
8K	0.183	3.82E-06	0.133	3.44E-06	(6,6)	9.82%	0.142	3.44E-06	(4,4)	(2,2)	10.03%
16K	0.183	7.64E-06	0.133	6.89E-06	(6,6)	9.79%	0.133	6.89E-06	(6,6)	(0,0)	9.79%
			Average Improvement(%)			9.53%	Average Improvement(%)				13.44%

5.6 Conclusions

We presented a fast yield-analysis framework to analyze memory yield which includes a very efficient, optimal algorithm for fault-repair analysis. Using this framework, we performed the first yield-aware analysis of redundancy and stand-by supply voltage for sub-threshold SRAMs. We observed that our method analyzes yield 600% faster than the previous methods in low V_{dd} regions without sacrificing accuracy. Moreover, we proposed a new technique to estimate and reduce the total memory stand-by leakage by replacing low-DRV cells. Through our method, total memory leakage was reduced by about 14% without any yield loss.

Chapter 6

Low-Power Multiple-Bit Upset Tolerant Memory Optimization

In this chapter, we propose a framework for analyzing Soft Error Rates (SER) including Multiple-Bit Upsets (MBU). Then, using this framework, we optimize the soft error tolerant voltage (V_{tol}) and interleaving distance (ID) of low-power, error-tolerant memories. Experimental results show that the total power can be reduced by an average of 30.5% with V_{tol} optimization and an average of 40.9% by simultaneously considering V_{tol} and ID together when compared to worst-case design practices.

To analyze MBU effect on memory array and minimize power, we utilize a Monte Carlo framework to analyze the impact of V_{dd} and V_{th} variability on soft error rates (SER) of various size memories. We then use this method to demonstrate power reduction in error-tolerant memories by optimizing the error tolerant supply voltage (V_{tol}) and the interleaving distance (ID) simultaneously. Specifically, our major contributions are as follows:

- We propose the first SER analysis framework considering Multiple-Bit Upsets (MBU).

- We then demonstrate the use of this framework to perform SER-aware voltage scaling to reduce power.
- We then are the first to optimize the memory interleaving distance to create low-power MBU-aware memories.

The rest of this chapter proceeds as follows: In Section 6.1, we present pertinent background on soft error reliability and our framework considering MBU effect. In Section 6.2, we propose a soft error aware voltage scaling method. Section 6.3 presents a new method that optimizes the interleaving distance to reduce power. In Section 6.4, we present our experimental results and analysis. Finally, Section 6.5 concludes the chapter.

6.1 Transient Radiation Faults in Memory

6.1.1 Soft Error Rate (SER) Model

SER exhibits an exponential dependence on the critical charge Q_{crit} [30] as

$$SER_{cell} = N_{flux} \times CS \times exp(\frac{-Q_{crit}}{Q_s}) \tag{6.1}$$

where N_{flux} is the intensity of the flux and CS is the area of the cross-section of the node. Q_s is the collection efficiency which depends on the doping profile. Q_{crit} is the critical charge [69] defined as

$$Q_{crit} = \int_0^{\tau_{flip}} I_D dt = C_{node} V_{dd} + I_{restore} \tau_{flip} \tag{6.2}$$

where I_D is the current induced by spike pulse, C_{node} is the capacitance of a node, and τ_{flip} and current $I_{restore}$ are the restoration time and current, respectively.

6.1.2 Multi-Bit Upset (MBU) Model ($\widehat{SER}$)

Cells adjacent to a particle strike are also likely to be upset and cause a MBU. We propose a method to calculate the SER including this effect. Our approach uses a previous MBU model [46] which considers the spatial charge density. We assume that N_{flux} determines the induced charge density (C/m^2) and that the induced charge generates the current ($\delta V/\delta t$). This is true because the induced charge creates the current spike [47]. This also means that we can formulate a model for MBUs using SER_{cell} by including N_{flux}. Therefore, the soft error rate for a cell j, including multi-bit errors is calculated as

$$\widehat{SER_j} = \frac{1}{K} \sum_{i=0}^{NM-1} e^{-\frac{\|i-j\|^2}{2\sigma_d^2}} SER_i \tag{6.3}$$

where $i, j \in [0, NM-1]$ is the index of a cell in an $N \times M$ memory array and $\|i-j\|$ is the Euclidean distance between the two cells at i and j. K is a normalization constant and the term $e^{-\|i-j\|^2/2\sigma_d^2}$ describes a Gaussian model for soft error rate decay between the two cells at i and j. σ_d denotes the deviation in SER spatial location. The local SER, SER_i, is the soft error rate considering only direct particle strikes to a memory cell i. Using Equation (6.3), the MBU soft error rate, $\widehat{SER_j}$ of each memory cell includes nearby upsets and MBUs.

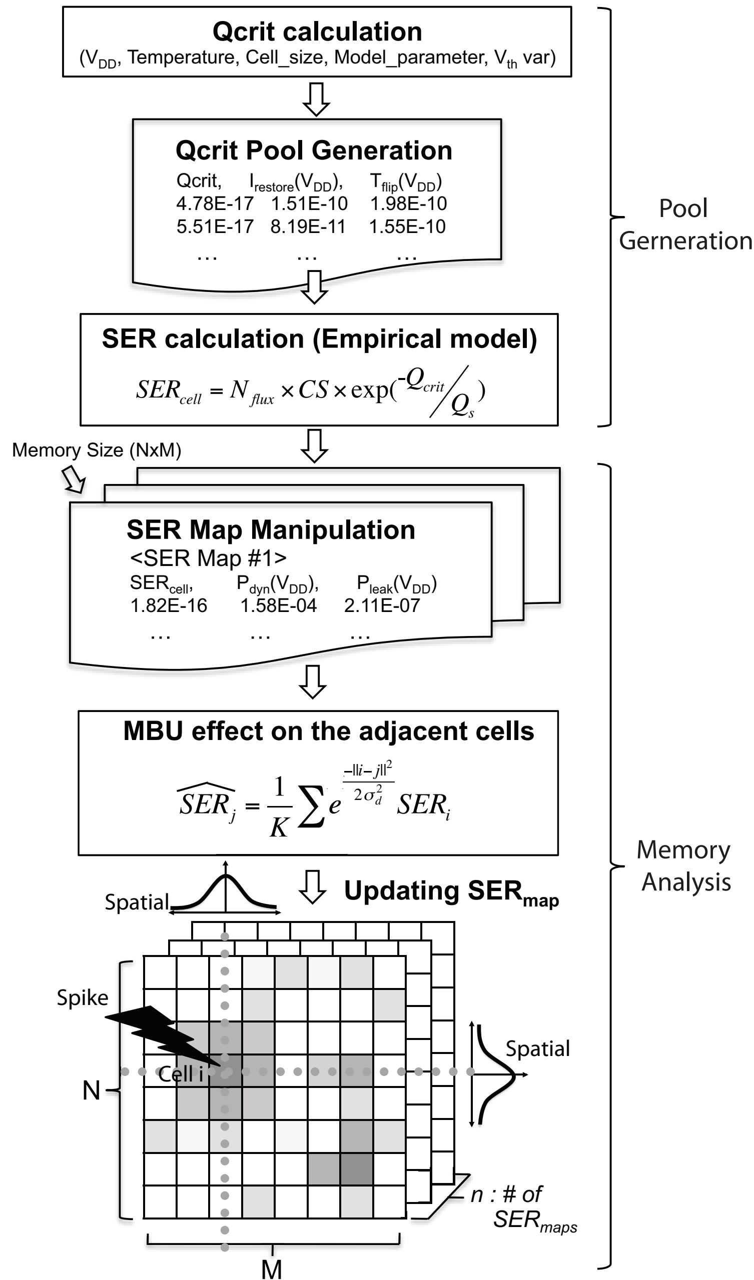


Figure 6.1: Monte Carlo Framework for SER_{map} manipulation

6.1.3 SER Analysis Framework

Our method analyzes the SER of an entire memory using Monte Carlo (MC) simulation and includes MBUs. An overview of our MC framework is shown in Figure 6.1. Our framework calculates the local SER_{cell} using each cell's critical charge Q_{crit} and V_{th} variation. It then generates a SER_{map} which is defined as a $N \times M$ array of memory cells.

As the first step, a pre-computed pool of individual cells is created to speed up simulation. Each cell's critical charge Q_{crit} is calculated using Equation (6.2) and includes $I_{restore}$, τ_{flip} by performing short MC simulation of variability parameter such as V_{th}. This step generates a large Q_{crit} pool with cells having the different $I_{restore}$ and τ_{flip}. Each cell's local SER, denoted by SER_{cell}, is calculated with the flux parameters using Equation (6.1).

We use random sampling from the pool data to perform MC analysis of entire memory arrays by creating n memory maps, SER_{map}. Each SER_{map} randomly selects NM cells from the previous pool (with replacement) to obtain each cell's local SER_i including variation. This pool also characterizes the SER of each cell SER_{cell}, cell dynamic power $P_{dyn}(V_{dd})$, and cell leakage power $P_{leak}(V_{dd})$ at a given supply voltage V_{dd} as a 3-tuple:

$$(SER_{cell}(V_{dd}), P_{dyn}(V_{dd}), P_{leak}(V_{dd})).$$

The last two terms will be used for power calculation in Section 6.3.

6.2 MBU Aware Voltage Scaling

According to Equation (6.1), SER_{cell} has an exponential dependency on V_{dd}. Reducing the supply voltage will lower the memory power, however, it will adversely affect

SER_{cell} as well. An important problem is to decide the best voltage while considering SER_{cell}. High V_{dd} wastes power but makes memory cells inherently more SEU tolerant. Low V_{dd} saves power, but requires other mechanisms such as ECC (which may consume power) to maintain SEU tolerance.

We define the soft error tolerant voltage V_{tol} and extend SER_{cell} to SER_{word}. SER_{word} is the soft error rate of a word not a single bit. Since ECC schemes such as Single Error Correction Double Error Detection (SEC-DED) [17] can fix a transient error per each word, our method calculates SER_{word} to get V_{tol} with SEC-DED. SER_{word} is determined with the word size W and each cell's SER_{cell} where $cell \in [0, W-1]$ as:

$$SER_{word}(V_{dd}) = \sum_{cell=0}^{W-1} SER_{cell}(V_{dd}). \tag{6.4}$$

Having defined SER_{word}, we can calculate the tolerant voltage V_{tol} and the worst-case voltage by sweeping V_{dd} for each SER_{map}. Since each cell's $SER_{cell}(V_{dd})$ is different, V_{dd} affects each SER_{word} depending on Equation (6.4). With different values of SER_{word}, V_{tol} can be determined when SER_{word} gives the probability including exactly one transient fault in the word. The worst-case voltage is determined similarly when SER_{word} is near zero. Among the various V_{tol} that we calculate, our method takes the maximum V_{tol} to ensure reliability of the entire memory.

In voltage range ($V_{dd} \in [V_{tol}$, worst-case $V_{dd}]$), only one transient error is allowed, but ECC is able to repair the error. In this case, the power is determined by the array, ECC and parity bits. In the region above the worst-case V_{dd} ($V_{dd} >$ worst-case V_{dd}), memories don't need ECC and parity bits. In this case, the power is dominated by the array power

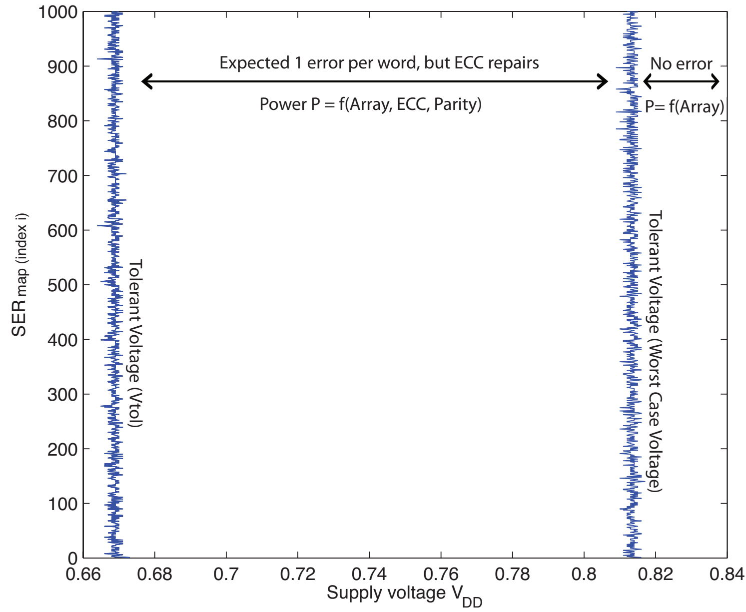


Figure 6.2: V_{tol} and worst-case voltage over n=1000 SER_{maps} at a normal flux [94]

consumption only.

Figure 6.2 shows the V_{tol} calculated for a normal flux condition [94] over the different 1000 SER_{map} memories. For example, in the voltage region (0.670V-0.813V), only one transient error is expected, but ECC is able to repair the error. In this case, the power is determined by the array, ECC and parity bits. In the region above 0.820V, memories don't need ECC and parity bits. In this case, the power is dominated by the array power consumption only.

We observe that the power consumption is different in each of these voltage regions. Moreover, V_{tol} is calculated assuming the maximum ID which is considered optimal by the previous works [5]. To reduce the power further, we propose to analyze the effect of ID on V_{tol} using the proper models. This will be further discussed in Section 6.3.

6.3 Power-Aware ID Optimization

We propose a method to select an optimized interleaving distance (OID) and V_{tol} simultaneously while considering memory SER and power consumption. Simply using the maximum ID as in [5] increases the power consumption because the word-line for the accessed row must enable all unselected cell's access transistors. This results in many "$half$" selected cells which increase the leakage power consumption and potentially disturb the read operation [33]. On the other hand, word-line segmenting[1] allows us to not half select these cells and significantly reduce the memory power consumption.

6.3.1 Interleaving Distance Power Model

Figure 6.3 shows an example of how ID affects power consumption of a row in various interleaved memory architectures (ID=1,2 and 4). Assume that the memory array consists of 16 bits with 4 4-bit words in each row. The radiation hits the first 4 bits causing a MBU. The maximum ID (ID=4) reduces SER_{word} by 1/4 times over the non-interleaved scheme (ID=1) due to the increased physical distance between the adjacent bits. This is the reason why many previous SER tolerant memory designs used the maximum ID.

The main disadvantage of the ID=4 case, however, is that the number of half-selected bit cells are increased. If the read operation accesses the bits $[A_0][A_1][A_2][A_3]$, we must enable the entire row during operation in the ID=4 case. This will half select the all other bits in the row and increase power consumption [33].

[1]Divided Word Line (DWL) : word-line is divided into plural blocks [88]

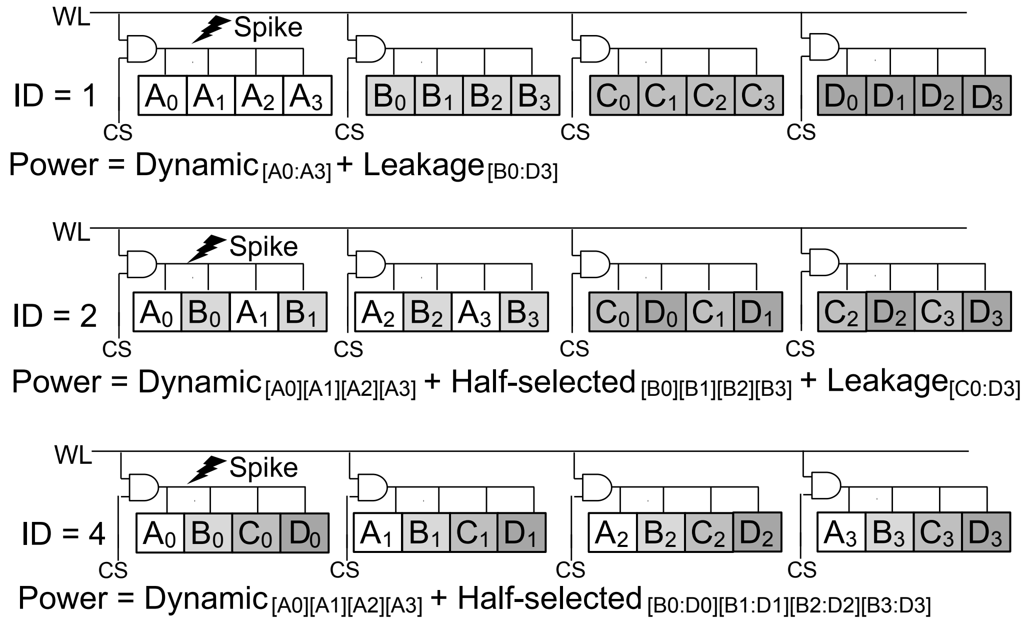


Figure 6.3: Example of ID effects (ID=1,2 and 4) on power in SRAM

Total memory power consumption with parity bits and ECC can be modeled by

$$P_{mem} = P_{array} + P_{parity} + P_{ECC}. \tag{6.5}$$

To analyze the impact of ID on the array power P_{array} at tolerant voltage V_{tol}, denoted by $P_{array}(V_{tol})$, we first define the power consumption of a row P_{row} at V_{tol}, denoted by $P_{row}(V_{tol})$, as the sum of accessed bit's power $P_{dyn}(V_{tol})$, the half-selected bit's power $P_{half}(V_{tol})$ and the leakage power of unselected cells $P_{leak}(V_{tol})$. This can be summarized

analytically as

$$
\begin{aligned}
P_{row}(V_{tol}) &= W \cdot P_{dyn}(V_{tol}) + N_{half} \cdot P_{half}(V_{tol}) \\
&+ N_{leak} \cdot P_{leak}(V_{tol}) \\
&= W \cdot P_{dyn}(V_{tol}) + W(ID-1)P_{half}(V_{tol}) \\
&+ (M - W \cdot ID) \cdot P_{leak}(V_{tol}) \\
&= \sum_{i \in W} P_{cell_i} + \sum_{j \in N_{half}} P_{cell_j} + \sum_{k \in N_{leak}} P_{cell_k}.
\end{aligned} \tag{6.6}
$$

$P_{dyn}(V_{tol})$ and $P_{leak}(V_{tol})$ are pre-computed in the cell pool in Section 6.1.3. $P_{half}(V_{tol})$ is also calculated with the cell having a non-precharged bitline voltage using $P_{dyn}(V_{tol})$. Each component of the previous power formulation can be described with P_{cell} and correspond index i, j and k at the tolerant voltage V_{tol}. In addition, W means the wordsize and N_{half} indicates the number of half-selected cells in a row which can be computed using $N_{half} = W(ID-1)$ and $N_{leak} = M - W \cdot ID$.

Given the power of a single row, we can define the $N \times M$ array power $P_{array}(V_{tol})$. Since memory operation allows a single row to be accessed at a time, the total memory power can be calculated using $P_{row}(V_{tol})$ and the rest of array $(N-1)M \cdot P_{leak}(V_{tol})$ as in

$$
\begin{aligned}
P_{array}(V_{tol}) &= P_{row}(V_{tol}) + (N-1)M \cdot P_{leak}(V_{tol}) \\
&= P_{row}(V_{tol}) + \sum_{i \in (N-1)M} P_{cell_i}.
\end{aligned} \tag{6.7}
$$

The term $(N-1)M \cdot P_{leak}(V_{tol})$ is calculated using $\sum_{i \in (N-1)M} P_{cell_i}$ at the tolerant volt-

age V_{tol} because each memory cell has different leakage power consumption due to V_{th} variation.

After this, we can substitute Equation (6.7) into P_{array} of Equation (7.5). To get the P_{parity} and P_{ECC} at V_{tol}, our method uses $P_{leak}(V_{tol})$ and P=number of parity bits with the inequality $2^P - 1 - P \geq W$ [25]. We also include the power of simulated XOR gates for P_{ECC}.

With the above model, we can analyze the effect that ID has on memory power consumption. Furthermore, we can reduce the power by optimizing the ID selection as will be discussed in the next section.

6.3.2 Optimal Interleaving Distance (OID)

While maximum ID is ideal for soft error tolerance, it can dramatically increase the power consumption. This section attempts to compromise between the memory power consumption while still being tolerant of soft errors. For example, if we select a non-maximum ID value, SER increases. However, if we increase the voltage to be higher than V_{tol}, we can compensate for the increased SER efficiently. This reduces the number of half-selected bits compared to the case using the maximum ID. Because the divided word line reduces the number of half-selected cells in non maximum ID cases.

To analyze this, we consider all ID cases with ID-aware power model discussed in section 6.3.1 to figure out which ID selection gives the minimum power consumption. Algorithm 2 describes the power optimization process, starting with the maximum ID. Using $Max(ID) = M/W$ as the initial value in Line 2-3, we update $V_{tol} = UpdateV_{tol}(ID)$ in Line 7 because the different ID yields the different V_{tol}. Then, we analyze the power con-

Algorithm 2 $Optimize_Power(Max(ID), V_{tol})$

Require: $N \times M$ array, Word size W, V_{tol}
Ensure: Power reduction using OID, updated V_{tol}

```
1:  // Starting with maximum ID
2:  Max(ID) ⇐ M/W
3:  ID ⇐ Max(ID)
4:  Power ⇐ MAX_INT
5:  // Deciding optimal ID and V_tol
6:  while ID ≥ 1 do
7:      V_tol ⇐ UpdateV_tol(ID)
8:      // Checking power improvement
9:      if Power ≥ CalculatePower(ID, V_tol) then
10:         Power ⇐ CalculatePower(ID, V_tol)
11:     end if
12:     ID ⇐ ID/2
13: end while
14: Return Power
```

sumption using $CalculatePower(ID, V_{tol})$ in Line 9-10. The two functions $UpdateV_{tol}(ID)$ and $CalculatePower(ID, V_{tol})$ denote the two processes described in Section 6.2 and Section 6.3.1.

6.4 Experimental Results

Our method generates memories with random variation using a 16K SRAM cell pool. The simulations are based on the 45nm PTM technology models [4] with a temperature of 25°C. We assume that transistors have independent $\pm 15\%/3\sigma$ variation of the nominal V_{th}. The pull-up/pull-down SRAM transistor width size ratio is 0.5 and $\frac{PR}{CR}=\frac{90nm/45nm}{180nm/45nm}$ with same length [59]. We set the flux parameters to be N_{flux}=56.5$m^{-2}s^{-1}$ which is typical particle flux [94], CS=0.296μm^2 [86], and altitude=1000ft. The simulation framework is written in Python and executed on an Ubuntu 10.04 system with 2.2GHz CPU and 4GB memory.

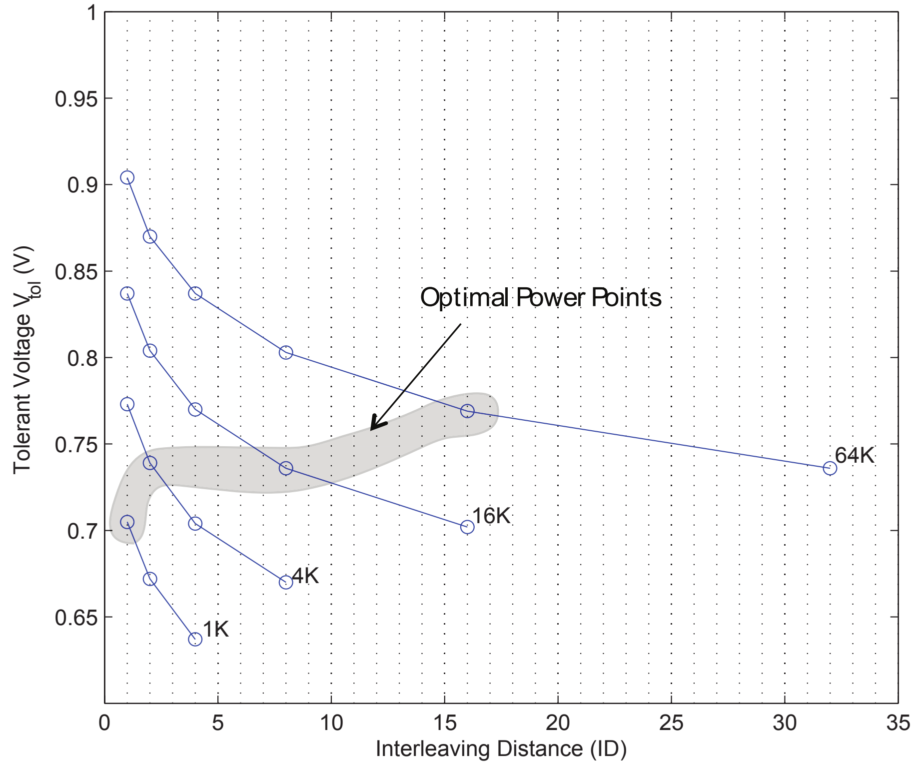


Figure 6.4: ID vs. V_{tol} and power optimal condition in 1K, 4K, 16K and 64K SRAMs with Flux1

6.4.1 Interleaving Distance Effect on V_{tol}

Figure 6.4 shows the ID effect on V_{tol} when memories are tolerant to the given flux. We compare various SRAM sizes (1K, 4K, 16K and 64K) that use SEC-DED. The figure shows that the memory having the maximum ID reduces the tolerant voltage V_{tol} in each memory size. Although the maximum ID reduces V_{tol}, it does not provide the lowest power memory. For example, we observe that the power optimal conditions (ID, V_{tol}) are not located at the maximum ID in Figure 6.4.

In addition, the V_{tol} of small memories is more sensitive to ID. For example, the ID's change (from $MaxID$=4 to ID=2) in 1K SRAM affects V_{tol}'s change more than the

ID's change (from $MaxID$=32 to ID=16) in 64K SRAM.

Table 6.1: Comparison of Power Reduction Effect with various Radiations (flux1, flux2 and flux4) in various SRAM sizes (W=8)

	SRAM without any ECC schemes and the worst-case supply voltage			Proposed method (V_{tol} optimization only)				Proposed method (V_{tol}, ID optimization both)			
Size	$V_{dd}(V)$	$MaxID$	$Power(W)$	$V_{tol}(V)$	$MaxID$	$Power(W)$	Improve(%)	$V_{tol}(V)$	OID	$Power(W)$	Improve(%)
1K	0.780	4	1.53E-03	**0.637**	4	1.02E-03	33.0%	0.705	1	**6.86E-04**	55.1%
4K	0.809	8	4.44E-03	**0.670**	8	3.04E-03	31.5%	0.739	2	**2.46E-03**	44.6%
16K	0.848	16	1.49E-02	**0.702**	16	1.02E-02	31.4%	0.736	8	**9.62E-03**	35.6%
64K	0.880	32	5.47E-02	**0.736**	32	3.82E-02	30.2%	0.769	16	**3.80E-02**	30.5%
$Flux1 = 56.5m^{-2}s^{-1}$				Average Improvement(%)			31.5%	Average Improvement(%)			41.4%
1K	0.815	4	1.68E-03	**0.667**	4	1.13E-03	32.8%	0.734	1	**7.56E-04**	55.1%
4K	0.844	8	4.84E-03	**0.701**	8	3.34E-03	31.0%	0.769	2	**2.67E-03**	44.9%
16K	0.881	16	1.61E-02	**0.737**	16	1.13E-02	30.2%	0.805	4	**1.06E-02**	34.6%
64K	0.914	32	5.90E-02	**0.770**	32	4.19E-02	29.0%	0.804	16	**4.18E-02**	29.1%
$Flux2 = 2 \times 56.5m^{-2}s^{-1}$				Average Improvement(%)			30.8%	Average Improvement(%)			40.9%
1K	0.846	4	1.79E-03	**0.702**	4	1.24E-03	31.0%	0.774	1	**8.29E-04**	53.7%
4K	0.881	8	5.25E-03	**0.737**	8	3.69E-03	29.7%	0.840	1	**2.92E-03**	44.3%
16K	0.915	16	1.74E-02	**0.768**	16	1.25E-02	28.4%	0.835	4	**1.13E-02**	34.8%
64K	0.948	32	6.35E-02	**0.804**	32	4.57E-02	28.0%	0.838	16	**4.54E-02**	28.6%
$Flux4 = 4 \times 56.5m^{-2}s^{-1}$				Average Improvement(%)			29.3%	Average Improvement(%)			40.4%

6.4.2 Power Reduction

Table 6.1 shows the power reduction using V_{tol} and the optimized ID on a variety of SRAM sizes when we assume three kinds of flux (flux1, flux2, and flux4). We compared two optimization strategies: 1) our proposed method with supply voltage reduction only (Section 6.2) and 2) our proposed method with both supply voltage reduction and ID optimization (Section 6.3). The baseline is an SRAM without any ECC schemes and the worst-case supply voltage for the soft error tolerance.

According to Table 6.1, supply voltage optimization with $MaxID$ can reduce the power consumption by an average of 30.5% compared to the worst-case supply voltage. Moreover, we are also able to reduce the power consumption by an average of 40.9% compared to the worst-case supply voltage if we use the OID with V_{tol}. This additional power optimization method is more efficient in small memory sizes (1-4K). The result also shows that the minimum supply voltage V_{tol} doesn't mean the minimum power consumption even though supply voltage optimization reduces the voltage V_{tol} successfully. In all cases (flux1, flux2, and flux4), 1K SRAM's V_{tol} (0.637V, 0.667V and 0.702V) is smaller than any other case. However, the power consumption is not the minimum. This is because the half-selected cells increase the power consumption at $MaxID$.

We also observe that large memories such as 16K and 64K don't have much power reduction using ID optimization. This implies that a large ID promises less power and strong immunity to the soft error in large memories.

6.5 Conclusions

We presented a soft error analysis framework to analyze memory SER. The framework considers MBU error locality as well as V_{th} variation. Using this framework, we demonstrated optimization of the soft error tolerant supply voltage V_{tol}. Moreover, we proposed a new method to estimate and reduce the total memory power by optimizing ID and V_{tol} simultaneously. Through our method, total memory power was reduced by about 40.9% while not sacrificing transient error tolerance.

Chapter 7

Dynamic Voltage Scaling for SEU-Tolerance in Low-Power Memories

Reliability issues are increasingly problematic as technology scales down and the supply voltage is lowered. Specifically, the Soft-Error Rate (SER) increases due to the reduced feature size and the reduced charge. This chapter describes an adaptive method to lower memory power using a dual V_{dd} in a column-based V_{dd} memory with Built-In Current Sensors (BICS). Using our method, we reduce the memory power by an average of 39.5% and increase the error immunity of the memory without the significant power overhead as in previous methods.

Researchers often use an empirical model for SER based on the critical charge Q_{crit} [30], but both the environmental and critical charge parameters of this model are challenging to estimate due to technology scaling and process variation. Other prior works [48] have proposed to use real measured data from radiation chambers to increase the accuracy of the prior models. This method improves the error rate accuracy, but it is costly in terms

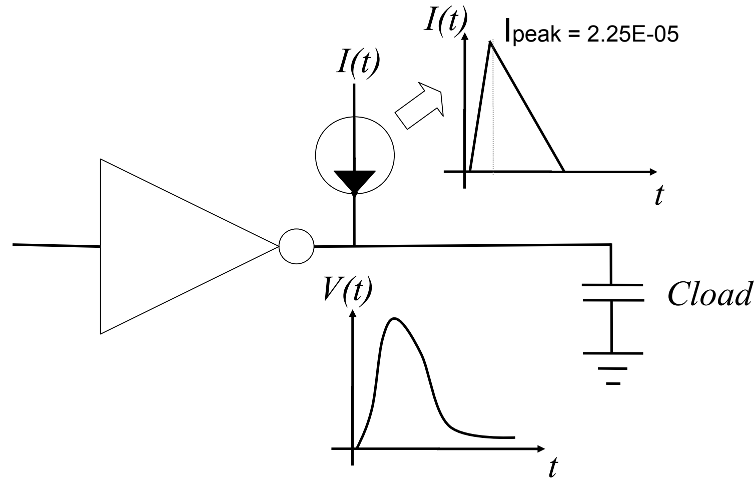


Figure 7.1: Gate-level SEU simulation methodology are used to analyze circuit robustness.

of resources and time to properly calibrate the model for the each chip designed.

The major issue with the prior approaches is that they can't dynamically react to immediate changes in the flux energy level. Built-In Current Sensors (BICS) have been proposed that detect transient errors in real time [56,67], so that the errors may be immediately detected and corrected by ECC. This enables the SER to be controlled within a tolerant range while the memory operates. Although it keeps the SER within acceptable margins, BICS and ECC increases the cost and power consumption of the chip.

In this work, we propose an SRAM architecture using BICS as feedback to detect the SEU and improve dynamic noise immunity using a dual-supply Dynamic Voltage Scaling (DVS) scheme. We implement this using a column-based V_{dd} array architecture with BICS. We then study the optimal supply voltage levels in terms of SER and power by

applying the new dual supply voltage scheme. Specifically, our major contributions are as follows:

- We propose a feedback system using BICS in column-based V_{dd} memories using DVS.
- We use a Monte Carlo framework to calculate the optimal voltage
- Our work is also the first to explicitly consider the Dynamic Noise Margin (DNM) of the memory cells.

The remainder of the chapter is organized as follows: Section 7.1 describes the overview of our BICS architecture. Section 7.2 introduces our MC framework and calculates the optimal voltage levels (V_{high} and DRV). Section 7.3 describes an power model using the dual V_{dd}. Then, Section 7.4 and 7.5 show our experimental setup and results, respectively, while Section 7.6 concludes the chapter.

7.1 Soft-Error Immunity

7.1.1 Built-In Current Sensor (BICS)

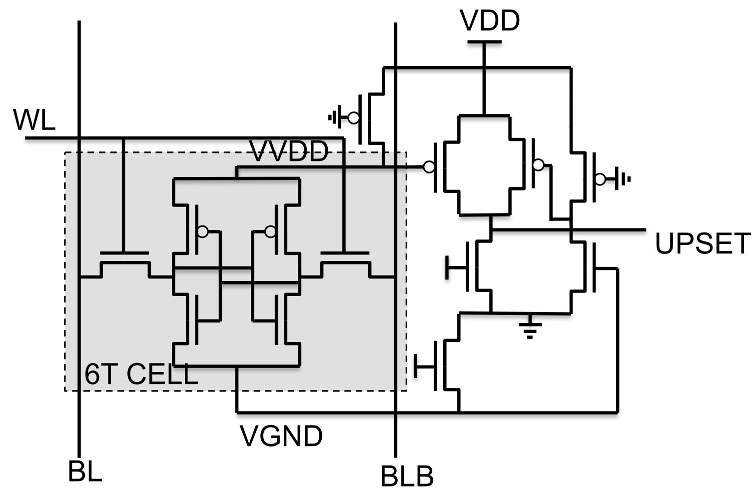
Figure 7.2(a) shows a BICS implemented alongside a 6T SRAM cell [56, 67, 78]. The BICS connects to each column at the bottom of the array. When a particle strikes an internal node of any memory cell in the column, the voltage of the internal node fluctuates due to the electron-hole pairs and immediately decreases the virtual V_{dd} (*VVDD*) of the BICS. This turns on the PMOS transistor in pull-up path of the BICS which asserts the *UPSET* signal to indicate the presence of a transient particle. In our approach, a Built-In Current Sensor (BICS) [78] detects transient particles and asserts a signal *UPSET* if a

transient particle is present. The *UPSET* signal selects a higher voltage in the column-based V_{dd} to assist recovering from a potential transient error, but allows stand-by operation at a lower supply voltage to save power.

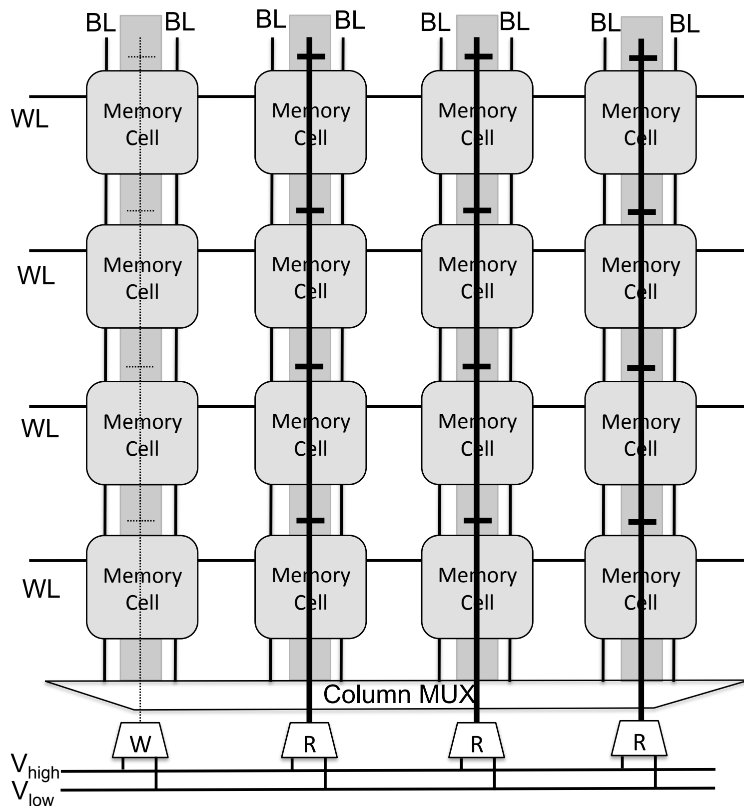
7.1.2 Column-based V_{dd} Memory

Column-based V_{dd} memories have been recently proposed to reduce power consumption [27, 92]. An example is shown in Figure 7.2(b) where each memory cell's V_{dd} is connected to the global V_{dd} in each column. Since SRAM read operations need higher V_{dd} for improved noise margins compared to write operations, a dual supply voltage saves power without performance or reliability degradation. When a column is read, the supply voltage is set to V_{high} and when a column is written, it is set to V_{low}. This approach reduces power by minimizing the supply voltage depending on the read/write operating pattern.

While column-based V_{dd} memory architectures have been used for power reduction during read and write operations, our approach assumes the same voltage level for both operations (read and write). Instead, we combine the BICS and column-based V_{dd} array to dynamically select the supply voltage in background data retention mode according to soft errors and operation as shown in Figure 7.3. The combination of these two techniques lowers power consumption by reducing the typical guardband voltage and improves fault tolerance.



(a) Built-In Current Sensors (BICS) detect particle strikes by monitoring the virtual supply and ground.



(b) The column-based V_{dd} enables BICS monitoring and supply selection of individual memory columns.

Figure 7.2: Previous works have separately used Built-In Current Sensor (BICS) for error detection and Column-based V_{dd} Array for dynamic power savings depending on the operation (read or write).

7.2 Adaptive Soft-Error Tolerance

7.2.1 Adaptive Supply Voltage Strategy

Our basic strategy is to use a low V_{dd} in stand-by operation and a V_{high} for active operation. Because memories spend most of the time in stand-by mode, a low V_{dd} can reduce the stand-by leakage power efficiently but as we previously saw a low V_{dd} reduces robustness.

For memory cells in non-accessed columns, V_{dd} of the column is adjusted to V_{high} when a SEU is detected. The low V_{dd} could be DRV, for example, but we need a method to improve DRV robustness. Figure 7.3 describes the architecture using V_{high} and DRV.

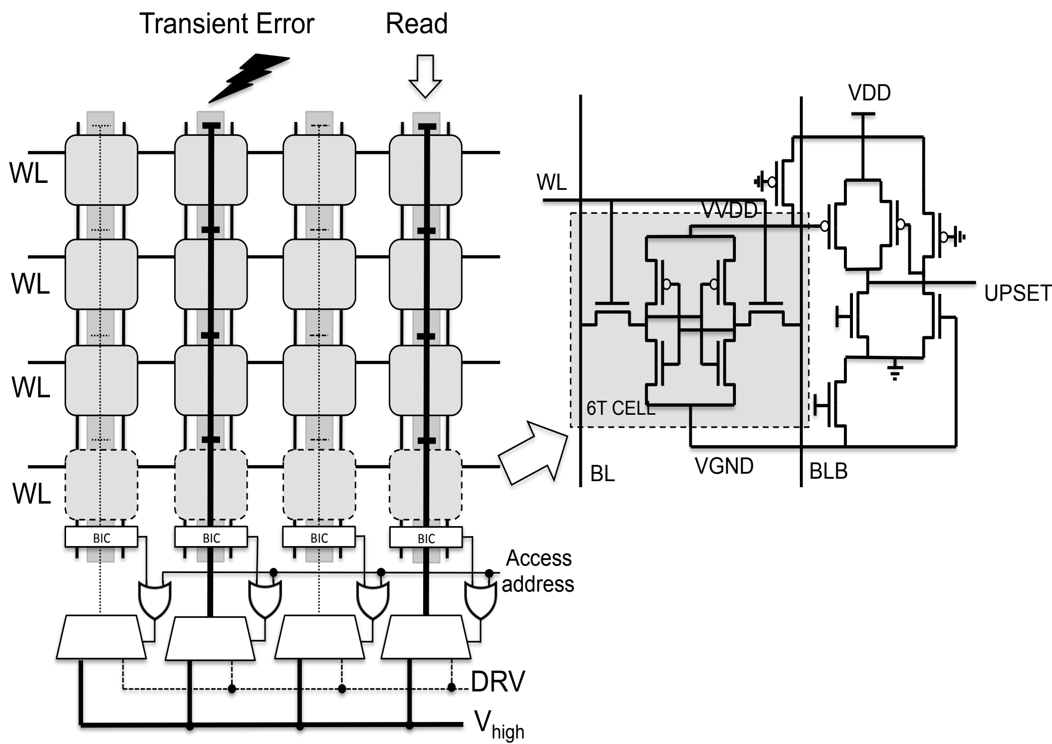


Figure 7.3: Our approach uses Built-In Current Sensor (BICS) together with a Column-based V_{dd} Array to detect SEUs at a column granularity.

V_{high} is only enabled through MUX when (1) the SEU occurs or (2) the column address (read/write) is addressed. To do this, we add one *OR* gate to the bottom of each column and connect *UPSET* signal and Column Selection (*CS*) signal as inputs. For example, if the SEU occurs in column 2 and the *CS* signal for read operation accesses column 4, only two columns are connected to V_{high}. The rest of columns are still connected to the low V_{dd}.

Our method adjusts V_{dd} of each column depending on whether a SEU is located in a column according to the BICS. This can happen in the background during idle periods. We assume that decoupling capacitors are used to prevent voltage fluctuation when the supply voltage switches between two voltages (V_{high} and DRV). Therefore, the power consumption is reduced by only using the high supply voltage when necessary. We calculate the V_{high} and DRV supply voltages by analyzing the memory access delay constraints of a read operation. Write operation is not directly considered, because the read operation is more critical than the write operation that needs less noise margin [92].

7.2.2 Memory Timing Access

Figure 7.4 shows our Monte Carlo framework that is used to analyze the impact of SEUs on memory timing. It uses several configuration parameters to specify the supply voltage, memory size, device parameters, and transistor variation. It then executes two independent processes. One process performs worst case delay characterization during normal memory operation while the other analyzes the recovery time when performing an access with V_{high} during a SEU. Both modules internally perform a voltage sweep to study the impact of V_{dd}.

The worst case delay is a quadratic function of the supply voltage with the coef-

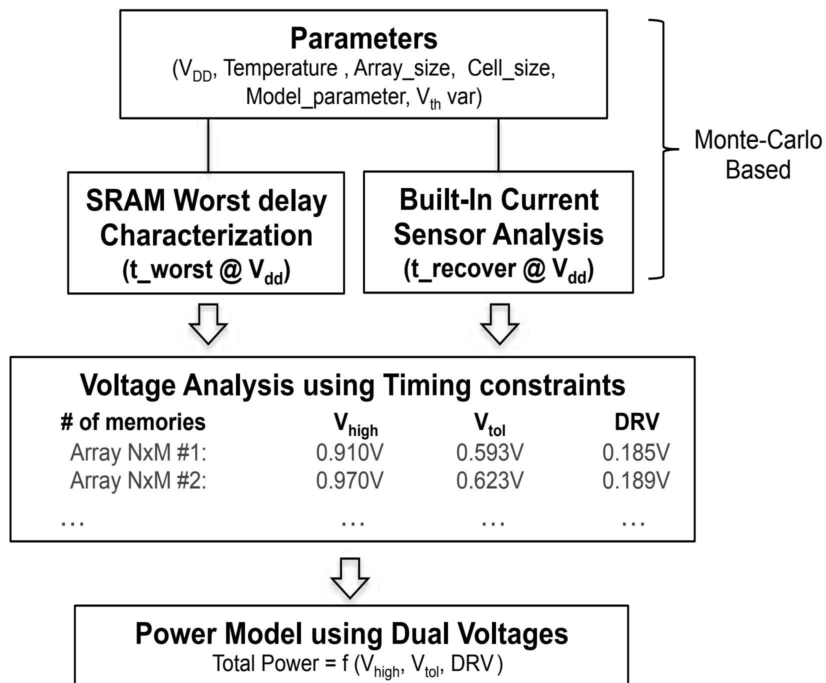


Figure 7.4: A Monte Carlo framework is used to analyze the timing and power of the low and high supply voltage levels.

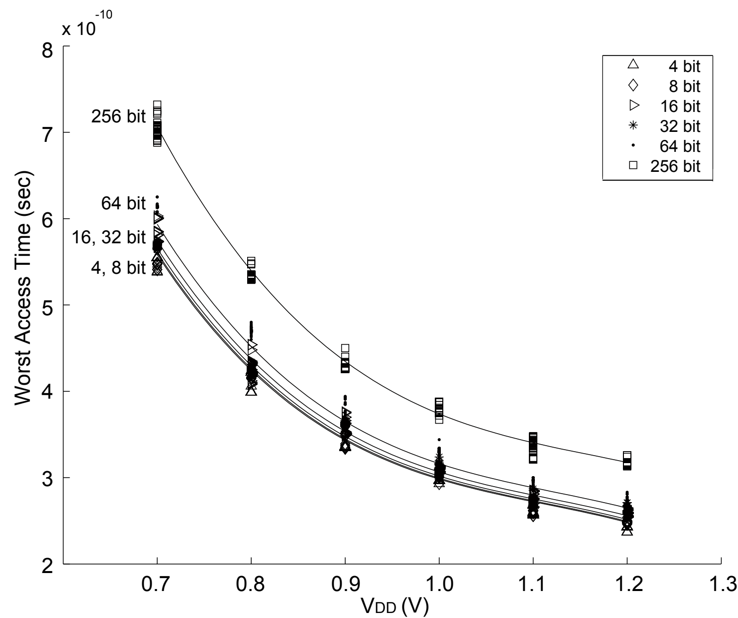


Figure 7.5: Memory worst case delay is fit to a non-linear model for various array sizes.

ficient depending on the array size,

$$t_{worst}(V_{dd}) = f(V_{dd}, M, N). \tag{7.1}$$

Figure 7.5 shows this using simulation data (V_{dd} and array size $N \times M$). Similarly, the recovery time from a SEU using the BICS architecture is measured as the time required for a memory node voltage to fully recover (99.9% of V_{dd}) using the dual voltage. This is a function of the memory column height due to the bit-line and supply rail capacitance and the supply voltage due to the memory cell drive strength,

$$t_{recover}(V_{dd}) = f(V_{dd}, DRV, N). \tag{7.2}$$

Figure 7.6 shows the recovery time $t_{recover}$ depending on column height N. Our method sets the supply voltages to the dual V_{dd}=0.4V and V_{dd}=1.2V. As expected, large column height N increases $t_{recover}$ in both cases (I_{peak}=2.25E-05 and I_{peak}=6.25E-05) due to the linear increase in capacitance.

7.2.3 Minimum Recovery Voltage V_{high}

Figure 7.7 illustrates our method's effectiveness on repair SEUs. In all plots, the x-axis is the transient time and the y-axis is the supply V_{dd}, the *UPSET* signal from the BICS and the internal cell node voltages. When a particle hits a memory cell operating at a low voltage, the *UPSET* signal is generated some time later (t_{BICS}). But the low V_{dd} driving strength can't make the cell recover from SEU so the SEU flips the memory cell (Case (A)). In Case (B), when the dual voltages are applied to the memory cell, the cell

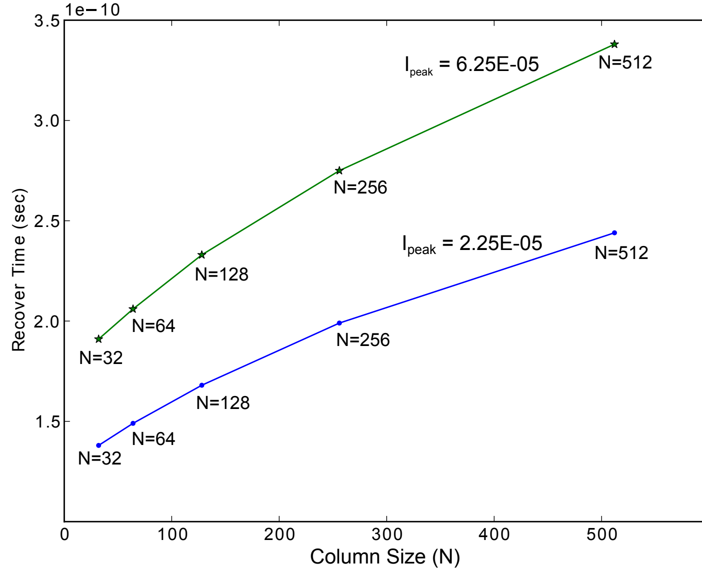


Figure 7.6: Plot (Column size N vs. $t_{recover}$) in different I_{peak}. Recovery time of memory array increases linearly as column size N increases.

recovers quickly after *UPSET* is generated. This means that the memory cell is corrected to the original value of the node even though the SEU occurs.

V_{high} must be large enough to prevent transient errors, but it should be set at a low value to preserve power. Granting that low V_{high} can reduce the power, too low V_{high} reduces the transistor's driving strength so that it causes read violation errors if it is insufficient low. To calculate a proper value of V_{high}, our method considers the recover time $t_{recover}$ of a memory cell and the worst case delay t_{worst} without a SEU as a constraint. In our feedback architecture, the *UPSET* signal is fed to a MUX to adjust the voltage to V_{high}, t_{MUX} is the time for multiplexing the V_{dd} through the MUX so that node voltage can be eventually recovered when the SEU occurs. Even after the supply voltage is adjusted to V_{high}, time is needed to increase the internal node voltage using V_{high} to recover.

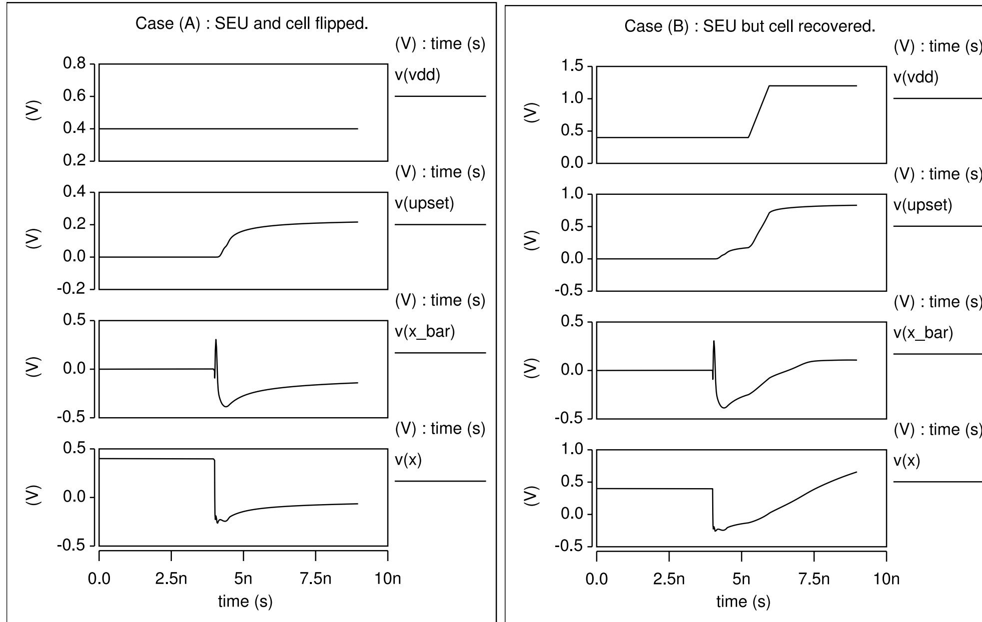


Figure 7.7: Simulation results using our feedback (BICS and dual V_{dd}) on a 1024 bit memory (32 cells in a column). Case (A): SEU flips memory cell, Case (B): SEU but cell recovers due to a higher voltage.

The total recovery time $t_{recover}$ is calculated using $t_{recover}$ = t_{BICS}+t_{MUX}+t_{cell}. Two of the sub-components (t_{BICS}, t_{MUX}) depend on the column height N while t_{cell} is largely determined by the supply voltage and cell driving strength itself.

The timing relation between t_{worst} and $t_{recover}$ can be established this criterion.

Criterion 2 *If a memory cell has a larger recovery time ($t_{recover}$) than the worst delay (t_{worst}), then the memory cell can not recover from SEU.*

The criterion shows an important view. A proper V_{high} lower-bound must be calculated using two delay parameters ($t_{recover}$ and t_{worst}) at a given V_{dd}. In other words, the condition ($t_{recover} > t_{worst}$) causes transient errors. Therefore, we can formulate the condition that

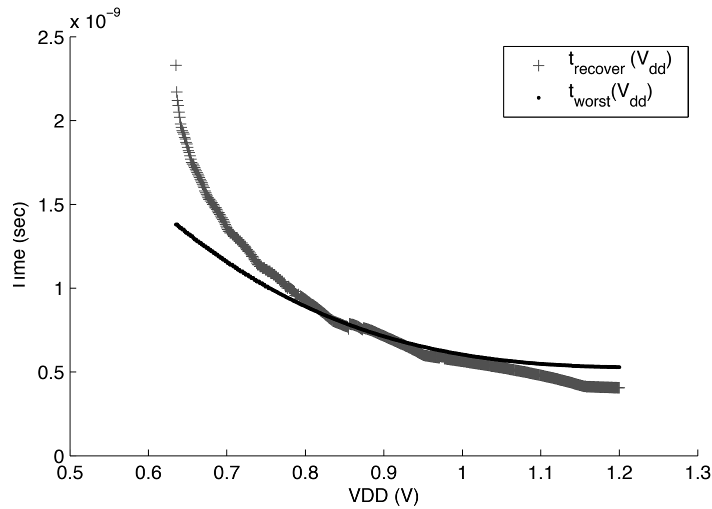


Figure 7.8: Calculation of V_{high} lower-bound using t_{worst} model and $t_{recover}$ simulation with I_{peak}=*3.25E-05* shows that the criterion is satisfied around $0.9V$ in 1024K SRAM

avoids the transient errors as:

$$t_{recover}(V_{dd}) \leq t_{worst}(V_{dd}). \tag{7.3}$$

V_{high} is the lowest V_{dd} that satisfies Equation (7.3) for an I_{peak} that we defined. For example, Figure 7.8 shows the plot of SRAM cell $t_{recover}$ and t_{worst} in various V_{high} voltages. Using this plot, the V_{high} lowerbound condition is satisfied near V_{dd}=$0.9V$. It is interesting to note that the quadratic coefficient of the recovery time is much less than the worst case memory. This is because the higher supply voltage enables the memory cell to more quickly recover from a SEU.

7.3 Probabilistic Power Model

Our architecture employs a dual voltage (V_{high} and DRV) selectively depending on the SEU occurrence and active operation frequency. This means that the V_{high} duration time differs depending on the circumstances (e.g. altitude and location) due to the flux of SEUs. This can be modeled probabilistically to estimate overall memory power.

7.3.1 Power Model using V_{high} and DRV

There are several components that must be considered to compute the power of our proposed approach. First, the column-based architecture needs an additional MUX in each column to select the proper supply voltage level. Also, the BICS operates independently from read/write operations to detect transient errors. The total memory power considering these issues is estimated as

$$P_{memory} = P_{array} + P_{MUXs} + P_{BICS}. \tag{7.4}$$

P_{array} is the $N \times M$ array power and denoted as $P_{array}(V_{high}, DRV)$ using a cell power P_{cell} and a ratio *p* and *(1-p)*. $p \in [0, 1]$ means the ratio of V_{high} duration time over total transient time. Inversely, *(1-p)* means the ratio of DRV duration over total transient time.

P_{array} is calculated using one of the following approaches: In one approach, we can see the dual V_{dd} effect in a traditional row-based array, applying V_{high} and DRV to an

entire array and estimate the power as:

$$\begin{aligned} P_{array} &= p \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} P_{cell(i,j)}(V_{high}) \\ &+ (1-p) \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} P_{cell(i,j)}(DRV). \end{aligned} \quad (7.5)$$

In another approach, we can apply V_{high} and DRV to columns selectively and estimate the power as:

$$\begin{aligned} P_{array} &= p \cdot \{P_{col}(V_{high}) + (M-1) \cdot P_{col}(DRV)\} \\ &+ (1-p) \cdot M \cdot P_{col}(DRV). \end{aligned} \quad (7.6)$$

In Equation (7.6), P_{col} shows the power consumption of a column based on $P_{col} = \sum_{i=1}^{N} P_{cell}(i,j)$ assuming 1 bit word size. Since the memory array consists of multiple bit words, Equation (7.6) uses the word size W to estimate the array power according to:

$$\begin{aligned} P_{array} &= p \cdot \{P_{col}(V_{high}) \cdot W + (M-W) \cdot P_{col}(DRV)\} \\ &+ (1-p) \cdot M \cdot P_{col}(DRV). \end{aligned} \quad (7.7)$$

In order to consider the power overhead of the supply P_{MUX} and P_{BICS} sensor, we simulate each component using the dual voltage stimulus with probabilities p and $(1-p)$ of SEUs occurring and sum up the respective power based on the corresponding memory size to calculate the overall power.

7.4 Experimental Setup

All simulations use the 45nm PTM technology models [4] with a temperature of 25°C. We assume that transistors have independent $\pm 15\%/3\sigma$ variation of the nominal V_{th}. The pull-up/pull-down SRAM transistor width size ratio is 0.5 and $\frac{PR}{CR}$=$\frac{90nm/45nm}{180nm/45nm}$ with identical gate lengths [59]. We set the maximum particle flux to be N_{flux}=56.5$m^{-2}s^{-1}$ [94] while the cross-sectional area is assumed to be CS=0.296μm^2 [86]. We generate memories ranging from 1K-256K using a memory compiler and then calculate the worst access delay using HSpice simulation. The worst case delay model t_{worst} model is fit using the Matlab command *nlinfit* due to the large t_{worst} simulation time on large memory arrays.

Our results are compared to a typical guardbanded approach as previously done in Chapter 6. The transient error tolerant voltage V_{tol} is selected such that no transient errors are expected with the given maximum particle flux.

7.5 Experimental Results

7.5.1 Dynamic Noise Margin (DNM) for SEU Analysis

Our method analyzes the Dynamic Noise Margin (DNM) during an SEU. We propose a method to analyze DNM for memories that use dual V_{dd} for power reduction. Figure 7.9 shows a plot with the x-axis representing the induced peak current I_{peak} and the y-axis as recovery time $t_{recover}$. Figure 7.9 shows 3 cases using dual V_{dd} (0.4V/0.9V, 0.4V/1.2V, and 0.4V/1.5V). The vertical lines are failure points. The lines show the maximum induced noise that can be tolerated given a recovery time constraint.

Using the plot, we can conclude two things : (1) our method can analyze DNM

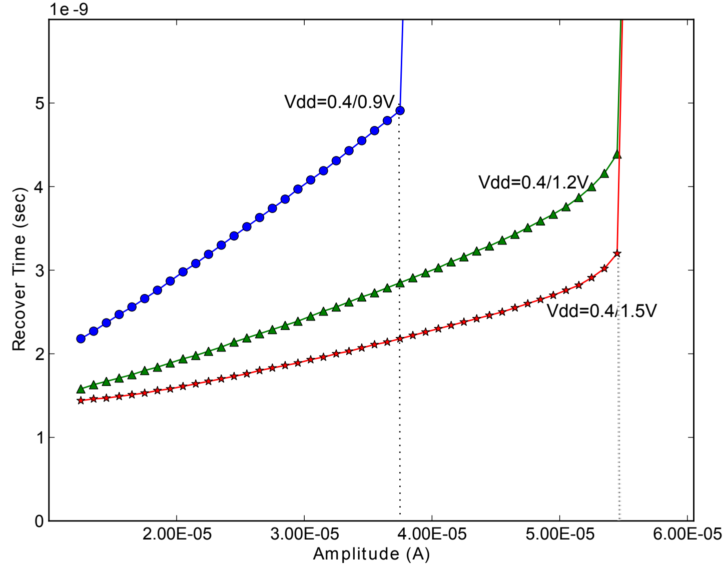


Figure 7.9: Peak current's amplitude (I_{peak}) vs. $t_{recover}$ in different dual V_{dd} combinations (1K SRAM) V_{high} determines the memory tolerance to a given I_{peak} amplitude and it should be calculated to optimal V_{dd} level to reduce the power.

when dual V_{dd} is used for recovering memory cell from SEU. (2) our method can analyze the optimal V_{dd} at given I_{peak} condition.

First, our method describes whether a SEU occurs the transient errors or not at given I_{peak} condition. This means that we can know how dual V_{dd} schemes are tolerant to given I_{peak}. For example, all three dual V_{dd} strategies can recover from a SEU at the condition (I_{peak}=2.25E-05) although $t_{recover}$ of V_{dd}=0.4V/0.9V is doubled compared to $t_{recover}$ of V_{dd}=0.4V/1.2V. However, at the condition (I_{peak}=3.75E-05), V_{dd}=0.4V/0.9V case fails to recover. This means that the DNM of the memory cell determines the tolerant amplitude I_{peak}=3.75E-05 in case (0.4V/0.9V). In other words, it is the maximum peak current that the memory cell tolerates with V_{dd}=0.4V/0.9V. Similarly, I_{peak}=5.45E-05 is the maximum peak current that the memory cell tolerates with V_{dd}=0.4V/1.2V and V_{dd}=0.4V/1.5V.

Second, our method determines the optimal V_{dd} that can tolerate a given noise I_{peak}. As expected, higher V_{dds} enable a faster recovery time. The recovery time $t_{recover}$ of the memory cell using V_{dd}=0.4V and 1.5V's is faster than the other cases at same I_{peak}. The higher V_{dd} increases the power unnecessarily although it enables the memory cell to recover quickly. For example, the both V_{high}=1.2V and V_{high}=1.5V cases have the same tolerance, however, the lower voltage should be selected to save power. For this reason, power-aware optimal V_{dd} should be near V_{high}=1.2V not 1.5V.

7.5.2 Power Reduction

We analyze the optimal supply voltage levels depending on the peak current I_{peak} that a flux generates [94]. The optimal voltages are calculated as V_{high}=0.948V, V_{tol}=0.607V at a flux N_{flux}=56.5$m^{-2}s^{-1}$ and DRV=0.186V. Table 7.1 shows the comparison of our two strategies: 1) our proposed method with V_{high} and DRV applied to entire array (column 3-column 4), b) our proposed method with V_{high} and DRV applied to selected columns (column 5-column 8). The baseline is a traditional SRAM with V_{tol} (column 2).

Table 7.1 and Table 7.2 compare proposed methods when energy particles strike the memory with probabilities p=0.1 and p=0.2, respective. We assume two cases since p value is not static and depends on the environment where the memory operates. It can be large number when the radiation particles strike frequently. According to Table 7.1, simply applying V_{high} and DRV to entire array can reduce the power consumption by an average of 16.38% compared to SRAM with static V_{tol} case. Applying V_{high} to the column with SEU and active columns selectively successfully reduces the power consumption by

an average of 55.09% compared to SRAM with V_{tol}. Also, when particles hit the memory more frequently (Table 7.2), the power reduction decreases to 7.03% and 49.87% compared to each case in Table 7.1. Because this method needs to use V_{high} two times more than Table 7.1 to avoid transient errors.

We also observe that our proposed method increases the power consumption in the case of small memories such as 1K, due to additional circuitry to implement the column-based V_{dd}. The additional circuit power overwhelms the small memory array power consumption, but in large memories (4K-256K), our method reduces the overall power efficiently in both scenarios.

Table 7.1: Power Reduction Results when Radiation strikes memory Once in various Sizes ($p = 0.1$)

Size	SRAM with V_{tol} only	Our Proposed I (V_{high},DRV to array)		Our Proposed II (V_{high},DRV to column)			
		Word size = 32		Word size = 32		Word size = 8	
	Power (W)	Power (W)	Improvement (%)	Power (W)	Improvement (%)	Power (W)	Improvement (%)
1K	3.336E-06	3.430E-06	-2.81%	2.767E-06	17.06%	2.431E-06	27.14%
4K	1.321E-05	1.115E-05	15.65%	6.963E-06	47.31%	6.293E-06	52.37%
16K	5.286E-05	3.883E-05	26.54%	1.944E-05	63.23%	1.810E-05	65.76%
64K	2.114E-04	1.363E-04	35.55%	5.836E-05	72.40%	5.572E-05	73.65%
256K	8.457E-04	5.448E-04	35.58%	2.075E-04	75.46%	2.022E-04	76.09%
	Avg. Improvement(%)		16.38%		55.09%		59.00%

Table 7.2: Power Reduction Results when Radiation strikes memory Twice in various Sizes ($p = 0.2$)

Size	SRAM with V_{tol} only	Our Proposed I (V_{high},DRV to array)		Our Proposed II (V_{high},DRV to column)			
		Word size = 32		Word size = 32		Word size = 8	
	Power (W)	Power (W)	Improvement (%)	Power (W)	Improvement (%)	Power (W)	Improvement (%)
1K	3.336E-06	3.744E-06	-12.23%	3.216E-06	3.61%	2.543E-06	23.78%
4K	1.321E-05	1.299E-05	1.68%	7.856E-06	40.55%	6.517E-06	50.69%
16K	5.286E-05	4.744E-05	10.25%	2.122E-05	59.86%	1.854E-05	64.92%
64K	2.114E-04	1.735E-04	17.92%	6.189E-05	70.73%	5.660E-05	73.23%
256K	8.457E-04	6.975E-04	17.53%	2.146E-04	74.62%	2.040E-04	75.88%
	Avg. Improvement(%)		7.03%		49.87%		57.70%

In both tables, we use 32 bits word size as the default, but we also analyze power consumption with an 8 bit word size. Smaller word sizes improve the power consumption, because our method only needs to enable a smaller number of columns during active read/write operations.

7.6 Conclusions

We presented a feedback system using BICS in column-based V_{dd} SRAM. We then used a Monte Carlo framework to calculate the optimal voltages (V_{high} and DRV) and demonstrated our method's power reduction to other approaches over a variety of memory sizes. In summary, our method is able to reduce the power by about 39.5% while not sacrificing transient error tolerance.

Chapter 8

Practical Issues with Fault Tolerant Low-power Memories

The application of fault tolerant methods to low-power memories has numerous practical issues which we discuss and present potential solutions in this chapter.

8.1 Measurement of Row/Column Leakage in Real Memory

Leakage current is usually measured as memory array in post-layout stage with silicon data. In a wafer where memory arrays designed, multiple probing nodes are connected to corresponding memory arrays. These nodes are located around each memory arrays so that test engineers probe the probing pad to measure leakage of the corresponding memory array using measurement equipment.

Instead of on-chip leakage power measurement, selecting a row and column that consumes more leakage current is a practical approach. Basically, this method uses different approaches for row and column because row replacement and column replacement require

different analysis. For a row replacement, current sensors can be used to detect a row consuming large leakage current among N rows of memory. In this approach, a current sensor is located at each row's separate power lines (V_{dd} and GND) so that I_{ds} can be monitored during a stand-by mode. For a column replacement, memory access time can be applied as a criterion to select the column consuming large leakage. Access time of every column is measured to analyze which column has large leakage current. A column that has a low access time is the column consuming large leakage based on the relation between I_{ds} and access time.

8.2 Fuse Architecture and Programming

Electrical fuses (eFuses) [68, 76, 77] have become an important technology to enable memory repair. An eFuse is a small, scalable solution that was introduced in the 0.25um technology and continues to be used in embedded system.

The structure of an eFuse is described in Figure 8.1. Programming is accomplished by controlling Electron Migration (EM) of the fuse link from the anode to the cathode node and activating NMOS transistor for a time t to create a current path at the same time. Specifically electrons migrate through silicide polysilicon lines when a high voltage is applied to anode through V_{source}. This silicide migration on the poly lines creates high resistance poly from previously conductive lines. The eFuse methodology increases the reliability of the circuits and improves the performance with a minimum cost.

However, it has a practical barrier in using high-K metal technology. As eFuse cuts the polysilicon link taking advantage of Joule heating, a high-K gate insulator using a

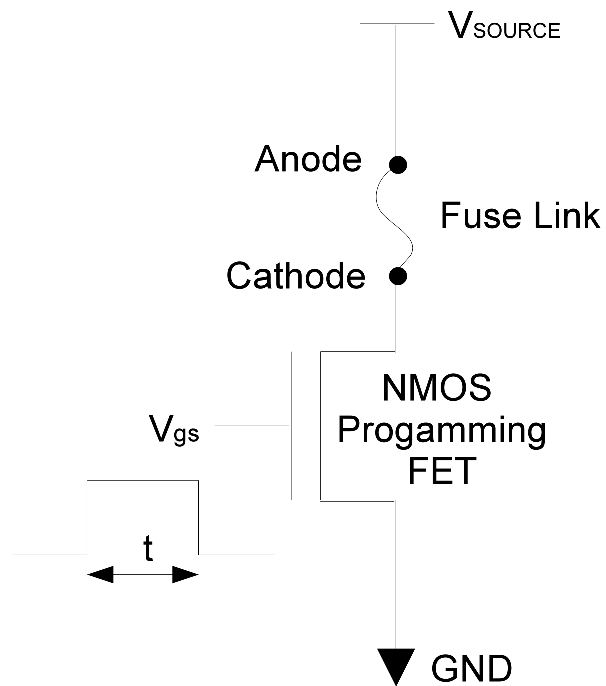


Figure 8.1: Electrical Fuse (eFuse) structure that is programmed by applying a high voltage during NMOS activation.

metal gate requires additional process and re-analysis of the high voltage isolating the link. This is because heat capacity of these materials (polysilicon and metal) are different from each other.

8.3 Generating Accurate Supply Voltages

Switched Capacitors (SC) are commonly used to convert a primary voltage to an internal accurate supply voltage. The basic idea of a SC is alternating the chip capacitance using switches to control the supply voltage based on an equation $Q = CV$. Switched capacitor converters also provide easy on-chip integration without large overhead and high efficiency in microwatt (uW) range.

However, accurate low voltage generation has become a practical issues due to reduced feature sizes and intolerance to noise. A small amount of noise can easily fluctuate the generated voltage due to the reduced driving strength. Transition time while switching the supply voltage is a practical issue, because, in a real circuit, the target supply voltage is generated smoothly in low voltage region.

This design challenge prevents the supply voltage from generating accurate supply voltage level and increased unexpected short circuit power. This problem can be resolved or avoided using design technique such as adding additional reference voltages with primary sources and increasing driver strength to get sharp transition.

8.4 Verification of SEUs/MBUs and SER

It is hard to verify the estimated SER with real occurrence rate of SEUs and MBUs even using real SEU data. This is because neutron, alpha flux density and radiation activity change continuously over the time and differ according to the location.

Real SER is calculated as the sum of neutron SER and alpha particle SER (SER = $SER_{neutron}$ + SER_{alpha}). However, measurement data is only the number of SEUs and MBUs during the test time. It can't separate the contribution of $SER_{neutron}$ and SER_{alpha}, which largely depend on the location and altitude. Researchers often use an Acceleration Factor (AF) to adjust the difference according to location and altitude where AF is measured in test chambers at different locations.

Specifically, to get realistic SER value, SER is measured at two test chamber locations (test facility at high altitude and in cave). The data measured from the facility at high altitude includes both contributions ($SER_{neutron}$ and SER_{alpha}). This is scaled to the corresponding altitude using AF and $SER_{neutron}$ difference between two location (high altitude and cave). The measurement gained from the facility in a cave is used to analyze only the SER_{alpha} contribution because neutron can't occur at low altitudes in a cave. As a final step, the scaled SER is added to pure SER_{alpha} to calculate a real SER.

Chapter 9

Conclusions and Future Work

Low-power operation is one of the important requirements in every design stage. Low V_{dd} enables the low-power operation, but it also complicates reliability issues due to the reduced noise margin and nano-scaled feature size. This thesis proposes practical methods to resolve the reliability and power issues together. Since the reliability and low-power issues are connected as we observed in Chapter 1, the solutions are carefully investigated and selected. This thesis proposes architectural/transistor level methods that consider the reliability and low power simultaneously for the first.

9.1 Thesis Contributions

The contributions of this thesis mainly focus on a methodology for low-power fault-tolerant memory design. Several methods are proposed that improve robustness, reliability and low-power memory design.

First, this thesis proposed a memory cell optimization. Using traditional 6T memory cell SNM analysis, the proposed method calculates the optimal transistor size improving

the cell power and reliability. This aspect can guide designers to set up the default transistor size or tweak the transistors size for their own purpose even when a new process is introduced.

Second, a method to improve memory yield while minimizing sub-threshold leakage power is presented. Specifically the thesis presented a self-repair algorithm that considers leakage reduction in sub-threshold memories by replacing rows and columns that have high leakage and large variability. Using this method, designers can be guided to calculate the optimal supply voltage and the required number of redundancies prior to chip tape-out. As expected, this method can reduce the leakage power of memory without sacrificing yield.

Third, a memory interleaving optimization method considering MBU is proposed to reduce power and improve SEU tolerance. Using this method, memory array designs can achieve fault tolerance and reduced power consumption. This method can be applied to devices that operates at a particular place or constant location.

Last, a dynamic method is proposed for SEU tolerance. This method improves the power using a built-in current sensor (BICS) and column V_{dd} at architecture level. Compared to the static methods, this method provides an adaptive solution when external noise/energy level changes and can be applied to mobile devices because devices that depends on the location of operation rather than using previous conservative guard-banding approaches.

9.2 Future Work

For the permanent fault tolerant design, this thesis employed eFuse structure to every leaf cell to disable the faulty cells or the leaky cells, or to enable the redundant cells. This work doesn't consider MOS-gated methods such as using NMOS/PMOS transistor as a switch instead of eFuse. Using MOS-gated transistor will reduce the design time, increase the reusability but it will change the power improvement because the CMOS switch consumes sub-threshold leakage power itself.

For the soft-error analysis, our results show that DNM is important and indicates a memory cell's tolerability to the external noise or radiation even when the input vector changes. Our work can be extended to calculate the critical energy level or the critical energy pulse width rather than just the peak noise current.

Bibliography

[1] A. Agarwal, Hai Li, and K. Roy. DRG-cache: a data retention gated-ground cache for low power. In *Design Automation Conference, 2002. Proceedings. 39th*, pages 473 – 478, 2002.

[2] A. Agarwal and K. Roy. A noise tolerant cache design to reduce gate and sub-threshold leakage in the nanometer regime. In *Low Power Electronics and Design, 2003. ISLPED '03. Proceedings of the 2003 International Symposium on*, pages 18 – 21, aug. 2003.

[3] K. Agarwal and S. Nassif. The impact of random device variation on SRAM cell stability in sub-90-nm CMOS technologies. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(1):86 –97, jan. 2008.

[4] ASU. Predictive Technology Model (PTM). `http://ptm.asu.edu`.

[5] Sanghyeon Baeg, ShiJie Wen, and R. Wong. SRAM interleaving distance selection with a soft error failure model. *Nuclear Science, IEEE Transactions on*, 56(4):2111 –2118, Aug. 2009.

[6] K. Baker and J. Van Beers. Shmoo plotting: the black art of ic testing. *Design Test of Computers, IEEE*, 14(3):90 –97, jul-sep 1997.

[7] S Bhardwaj, Wenping Wang, R Vattikonda, Yu Cao, and S Vrudhula. Predictive modeling of the NBTI effect for reliable design. *IEEE Custom Integrated Circuits Conference*, pages 189 – 192, Sep 2006.

[8] K. Bhattacharya and N. Ranganathan. RADJAM: A novel approach for reduction of soft errors in logic circuits. In *VLSI Design*, pages 453 –458, Jan. 2009.

[9] A Bhavnagarwala, X Tang, and J Meindl. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *IEEE Journal of Solid State Circuits*, pages 658–665, 2001.

[10] D. Blaauw, Steven M. Martin, Trevor N. Mudge, and Krisztián Flautner. Leakage current reduction in VLSI systems. *Journal of Circuits, Systems, and Computers*, 11(6):621–636, 2002.

[11] B Calhoun and A Chandrakasan. Analyzing static noise margin for sub-threshold SRAM in 65nm CMOS. *Proc. ESSCIRC*, Jan 2005.

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

[13] B Calhoun and A Chandrakasan. A 256-kb 65-nm sub-threshold SRAM design for ultra-low-voltage operation. *IEEE Journal of Solid State Circuits*, 2007.

[14] B Calhoun, F Honore, and A Chandrakasan. Design methodology for fine-grained

leakage control in MTCMOS. In *Low Power Electronics and Design, 2003. ISLPED '03. Proceedings of the 2003 International Symposium on*, pages 104 – 109, aug. 2003.

[15] V. Chandra and R. Aitken. Impact of technology and voltage scaling on the soft error susceptibility in nanoscale CMOS. In *Defect and Fault Tolerance of VLSI Systems*, pages 114 –122, Oct. 2008.

[16] A Chen. Redundancy in LSI memory array. *IEEE Journal of Solid State Circuits*, pages 291–293, Oct 1969.

[17] C. L. Chen and M. Y. Hsiao. Error-correcting codes for semiconductor memory applications: A state-of-the-art review. *IBM Journal of Research and Development*, 28(2):124 –134, March 1984.

[18] Q. Chen, A. Guha, and K. Roy. An accurate analytical snm modeling technique for srams based on butterworth filter function. In *VLSI Design, 2007. 20th International Conference on*, pages 615 –620, jan. 2007.

[19] Li Ding and P. Mazumder. Dynamic noise margin: definitions and model. In *VLSI Design, 2004. Proceedings. 17th International Conference on*, pages 1001 – 1006, 2004.

[20] P.E. Dodd, F.W. Sexton, and et al. Three-dimensional simulation of charge collection and multiple-bit upset in Si devices. *Nuclear Science, IEEE Transactions on*, 41(6):2005 –2017, Dec 1994.

[21] K Flautner, N Kim, S Martin, D Blaauw, and T Mudge. Drowsy caches: simple techniques for reducing leakage power. *Computer Architecture*, Jan 2002.

[22] L. B. Freeman. Critical charge calculations for a bipolar SRAM array. *IBM Journal of Research and Development*, 40:119–129, January 1996.

[23] Bo Fu and P. Ampadu. Leakage power minimization of nanoscale cmos circuits via non-critical path transistor sizing. In *Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference on*, pages 1101 –1104, dec. 2006.

[24] M. Goudarzi and T. Ishihara. Row/column redundancy to reduce sram leakage in presence of random within-die delay variation. In *Low Power Electronics and Design, 2008. ISLPED '08. Proceedings of the 2008 International Symposium on*, pages 93 –98, aug. 2008.

[25] R. W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 26(2):147–160, 1950.

[26] F Hamzaoglu, Y Te, A Keshavarzi, and K Zhang. Dual-Vt SRAM cells with full-swing single-ended bit line sensing for high-performance on-chip cache in 0.13 mu;m technology generation. In *Low Power Electronics and Design, 2000. ISLPED '00. Proceedings of the 2000 International Symposium on*, pages 15 – 19, Jan 2000.

[27] F Hamzaoglu, Yih Wang, P Kolar, Liqiong Wei, Yong-Gee Ng, U Bhattacharya, and K Zhang. Bit cell optimizations and circuit techniques for nanoscale sram design. *Design & Test of Computers, IEEE*, 28(1):22 – 31, 2011.

[28] S Hanson, M Seok, D Sylvester, and D Blaauw. Nanometer device scaling in subthreshold logic and SRAM. *IEEE TED*, pages 175–185, 2008.

[29] S. Hanson, D. Sylvester, and D. Blaauw. A new technique for jointly optimizing

gate sizing and supply voltage in ultra-low energy circuits. In *Low Power Electronics and Design, 2006. ISLPED'06. Proceedings of the 2006 International Symposium on*, pages 338 –341, oct. 2006.

[30] P Hazucha and C Svensson. Impact of CMOS technology scaling on the atmospheric neutron soft error rate. *Nuclear Science, IEEE Transactions on*, 47(6, Part 3):2586 – 2594, Dec 2000.

[31] R. Heald and P. Wang. Variability in sub-100nm SRAM designs. In *Computer Aided Design. ICCAD, 2004 IEEE/ACM International Conference on*, pages 347 – 352, nov. 2004.

[32] G.M. Huang, Wei Dong, Yenpo Ho, and Peng Li. Tracing sram separatrix for dynamic noise margin analysis under device mismatch. In *Behavioral Modeling and Simulation Workshop, 2007. BMAS 2007. IEEE International*, pages 6 –10, sept. 2007.

[33] R Joshi, R Kanj, and V Ramadurai. A novel column-decoupled 8T cell for low-power differential and domino-based SRAM design. *VLSI Systems, IEEE Transactions on*, PP(99):1 – 14, 2010.

[34] J. Kao, S. Narendra, and A. Chandrakasan. MTCMOS hierarchical sizing based on mutual exclusive discharge patterns. In *Design Automation Conference, 1998. Proceedings*, pages 495 –500, june 1998.

[35] T Kawagoe and J Ohtani et al. A built-in self-repair analyzer (CRESTA) for embedded DRAMs. In *ITC*, pages 567–574, 2000.

[36] Stefanos Kaxiras, Zhigang Hu, and Margaret Martonosi. Cache decay: exploiting

generational behavior to reduce cache leakage power. *ISCA '01: Proceedings of the 28th annual international symposium on Computer architecture*, Jun 2001.

[37] V. Khandelwal and A. Srivastava. Leakage control through fine-grained placement and sizing of sleep transistors. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(7):1246 –1255, july 2007.

[38] C Kim, J Kim, S Mukhopadhyay, and K Roy. A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(3):349–357, 2005.

[39] J Kim, M McCartney, K Mai, and B Falsafi. Modeling SRAM failure rates to enable fast, dense, low-power caches. *IEEE workshop on silicon errors in logic*, 2009.

[40] Nam Sung Kim, K. Flautner, D. Blaauw, and T. Mudge. Drowsy instruction caches. leakage power reduction using dynamic voltage scaling and cache sub-bank prediction. In *Microarchitecture, 2002. (MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on*, pages 219 – 230, 2002.

[41] Seokjoong Kim and M.R. Guthaus. Leakage-aware redundancy for reliable sub-threshold memories. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 435 –440, june 2011.

[42] Seokjoong Kim and M.R. Guthaus. Low-power multiple-bit upset tolerant memory optimization. In *Computer-Aided Design (ICCAD), 2011 IEEE/ACM International Conference on*, pages 577 –581, nov. 2011.

[43] Seokjoong Kim and M.R. Guthaus. SNM-aware power reduction and reliability

improvement in 45nm SRAMs. In *VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th International Conference on*, pages 204 –207, oct. 2011.

[44] Sy-Yen Kuo and W.K. Fuchs. Efficient spare allocation for reconfigurable arrays. *Design Test of Computers, IEEE*, 4(1):24 –31, feb. 1987.

[45] Chor Ping Low and Hon Wai Leong. Minimum fault coverage in memory arrays: a fast algorithm and probabilistic analysis. *IEEE Trans. on CAD of Integrated Circuits and Systems*, pages 681–690, June 1996.

[46] R. C. Martin, N. M. Ghoniem, and et al. The size effect of ion charge tracks on single event multiple-bit upset. *Nuclear Science, IEEE Transactions on*, 34(6):1305 –1309, Dec. 1987.

[47] G. C. Messenger. Collection of charge on junction nodes from ion tracks. *Nuclear Science, IEEE Transactions on*, 29(6):2024 –2031, Dec. 1982.

[48] S.E. Michalak, K.W. Harris, N.W. Hengartner, B.E. Takala, and S.A. Wender. Predicting the number of fatal soft errors in los alamos national laboratory's asc q supercomputer. *Device and Materials Reliability, IEEE Transactions on*, 5(3):329 – 335, sept. 2005.

[49] V Moalemi and A Afzali-Kusha. Subthreshold 1-bit full adder cells in sub-100 nm technologies. *IEEE Computer Society Annual Symposium on VLSI*, Jan 2007.

[50] S. Mukhopadhyay, K Kim, and et al. Self-repairing SRAM for reducing parametric failures in nanoscaled memory. In *VLSI Circuits Symposium*, pages 132–133, 2006.

[51] S. Mukhopadhyay, H. Mahmoodi, and K. Roy. Statistical design and optimization of SRAM cell for yield enhancement. In *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, pages 10 – 13, nov. 2004.

[52] 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*, 24(12):1859 – 1880, dec. 2005.

[53] 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*, 27(1):174 –183, jan. 2008.

[54] P. C. Murley and G. R. Srinivasan. Soft-error monte carlo modeling program, SEMM. *IBM Journal of Research and Development*, 40(1):109 –118, Jan. 1996.

[55] O. Musseau, F. Gardic, and et al. Analysis of multiple bit upsets (MBU) in CMOS SRAM. *Nuclear Science, IEEE Transactions on*, 43(6):2879 –2888, Dec 1996.

[56] E.H. Neto, I. Ribeiro, and et al. Using bulk built-in current sensors to detect soft errors. *Micro, IEEE*, 26(5):10 –18, Sept. 2006.

[57] E. Normand. Single event upset at ground level. *Nuclear Science, IEEE Transactions on*, 43(6):2742 –2750, Dec 1996.

[58] Philipp Ohler, Sybille Hellebrand, and Hans-Joachim Wunderlich. An integrated built-in test and repair approach for memories with 2D redundancy. In *ETS*, pages 91–96, May 2007.

[59] A Pavlov and M Sachdev. CMOS SRAM circuit design and parametric test in nano-scaled technologies: process-aware SRAM design. *Springer*, Jan 2008.

[60] M Pelgrom, A Duinmaijer, and A Welbers. Matching properties of MOS transistors. *IEEE Journal of Solid-State Circuits*, 24(5):1433–1440, Oct 1989.

[61] M. Powell, Se-Hyun Yang, B. Falsafi, K. Roy, and T.N. Vijaykumar. Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories. In *Low Power Electronics and Design, 2000. ISLPED '00. Proceedings of the 2000 International Symposium on*, pages 90 – 95, 2000.

[62] Huifang Qin, Yu Cao, D. Markovic, A. Vladimirescu, and J. Rabaey. SRAM leakage suppression by minimizing standby supply voltage. In *Quality Electronic Design, 2004. Proceedings. 5th International Symposium on*, pages 55 – 60, 2004.

[63] Huifang Qin, Yu Cao, D. Markovic, A. Vladimirescu, and J. Rabaey. Standby supply voltage minimization for deep sub-micron SRAM. *Microelectronics Journal*, Jan 2005.

[64] J Rabaey, A Chandrakasan, and B Nikolic. Digital integrated circuits: A design prospective. second edition. *Prentice Hall*, Jan 2003.

[65] R. Rajaraman, J. S. Kim, and et al. SEAT-LA: A soft error analysis tool for combinational logic. In *VLSI Design*, 2006.

[66] A Raychowdhury, S Mukhopadhyay, and K Roy. A feasibility study of subthreshold SRAM across technology generations. In *Computer Design: VLSI in Computers and*

Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference on, pages 417–422, oct. 2005.

[67] Pedro Reviriego, Juan Antonio Maestro, and Chris J. Bleakley. Reliability analysis of memories protected with BICS and a per-word parity bit. *ACM Trans. Des. Autom. Electron. Syst.*, 15:18:1–18:15, March 2010.

[68] N. Robson, J. Safran, C. Kothandaraman, A. Cestero, Xiang Chen, R. Rajeevakumar, A. Leslie, D. Moy, T. Kirihata, and S. Iyer. Electrically programmable fuse (efuse): From memory redundancy to autonomic chips. In *Custom Integrated Circuits Conference, 2007. CICC '07. IEEE*, pages 799 –804, sept. 2007.

[69] P Roche, J Palau, and et al. Determination of key parameters for SEU occurrence using 3-D full cell SRAM simulations. *Nuclear Science, IEEE Transactions on*, 46(6):1354 – 1362, Dec 1999.

[70] Erick Schonfeld. Android phones pass 700,000 activations per day, approaching 250 million total. *TechCruch*, Dec 2011.

[71] E Seevinck, F List, and J Lohstroh. Static-noise margin analysis of MOS SRAM cells. *IEEE Journal of Solid-State Circuits*, sc-22(5):748–754, Oct 1987.

[72] W Shi and WK Fuchs. Probabilistic analysis and algorithms for reconfiguration of memory arrays. *IEEE TCAD*, pages 1153–1160, Sep 1992.

[73] H. Soeleman, K. Roy, and B. Paul. Robust ultra-low power Sub-threshold DTMOS logic. In *Low Power Electronics and Design, 2000. ISLPED '00. Proceedings of the 2000 International Symposium on*, pages 25 – 30, 2000.

[74] D Sylvester, S Hanson, M Seok, Y Lin, and D Blaauw. Designing robust ultra-low power circuits. *IEEE International Electron Devices Meeting*, Jan 2008.

[75] R Tanabe, H Anzai, Y Ashizawa, and H Oka. The optimization of low power operation SRAM circuit for 32nm node. *Simulation of Semiconductor Processes and Devices*, 12:397–400, Sep 2007.

[76] C. Tian, B. Park, C. Kothandaraman, J. Safran, D. Kim, N. Robson, and S.S. Iyer. Reliability qualification of cosi2 electrical fuse for 90nm technology. In *Reliability Physics Symposium Proceedings, 2006. 44th Annual., IEEE International*, pages 392 –397, march 2006.

[77] W. Tonti. eFuse design and reliability. In *Integrated Reliability Workshop Final Report, 2008. IRW 2008. IEEE International*, page 145, oct. 2008.

[78] F. Vargas and M. Nicolaidis. SEU-tolerant SRAM design based on current monitoring. In *Proceedings of the 24th International Symposium on Fault-Tolerant Computing (FTCS)*, pages 106 – 115, Jun 1994.

[79] E.I. Vatajelu, A. Go andmez Pau, M. Renovell, and J. Figueras. Transient noise failures in sram cells: Dynamic noise margin metric. In *Test Symposium (ATS), 2011 20th Asian*, pages 413 –418, nov. 2011.

[80] N Verma and A Chandrakasan. Ultra low voltage SRAM design. *Embedded Memories for Nano-scale VLSIs*, Jan 2009.

[81] A Wang, A Chandrakasan, and et al. A 180-mV subthreshold FFT processor using a minimum energy design methodology. *IEEE Journal of Solid State Circuits*, 2005.

[82] J Wang and B Calhoun. Canary replica feedback for near-drv standby vdd scaling in a 90nm SRAM. *IEEE Custom Integrated Circuits Conference*, pages 29–32, Jan 2007.

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

[84] J Wang, A Singhee, R Rutenbar, and B Calhoun. Statistical modeling for the minimum standby supply voltage of a full SRAM array. *33rd European Solid State Circuits Conference*, Jan 2007.

[85] Kai-Chiang Wu and D. Marculescu. Power-aware soft error hardening via selective voltage scaling. In *Computer Design, 2008. ICCD 2008. IEEE International Conference on*, pages 301 –306, Oct. 2008.

[86] Fu-Liang Yang, Cheng-Chuan Huang, and et al. 45nm node planar-SOI technology with 0.296 μm^2 6T-SRAM cell. In *VLSI Technology, 2004. Digest of Technical Papers. 2004 Symposium on*, pages 8 – 9, June 2004.

[87] Fu-Liang Yang, Jiunn-Ren Hwang, and Yiming Li. Electrical characteristic fluctuations in sub-45nm CMOS devices. *IEEE Custom Integrated Circuits Conference*, pages 691–694, Sept 2006.

[88] M. Yoshimoto, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, S. Nagao, S. Kayano, and T. Nakano. A divided word-line structure in the static RAM and its application to a 64k full CMOS RAM. *Solid-State Circuits, IEEE Journal of*, 18(5):479 –485, oct. 1983.

[89] B Zhai, S Hanson, and et al. A variation-tolerant sub-200 mV 6-T subthreshold SRAM. *IEEE Journal of Solid State Circuits*, pages 2338–2348, 2008.

[90] B Zhai, L Nazhandali, J Olson, and A Reeves. A 2.60 pj/inst subthreshold sensor processor for optimal energy efficiency. *VLSI Circuits*, Jan 2006.

[91] Bin Zhang, A. Arapostathis, S. Nassif, and M. Orshansky. Analytical modeling of sram dynamic stability. In *Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on*, pages 315 –322, nov. 2006.

[92] Kevin Zhang, U. Bhattacharya, Zhanping Chen, and F Hamazaohlu. A 3-ghz 70-mb sram in 65-nm cmos technology with integrated column-based dynamic power supply. *Solid-State Circuits, IEEE Journal of*, 51(1):146 – 151, Jan 2006.

[93] Ming Zhang and N.R. Shanbhag. Soft-error-rate-analysis (SERA) methodology. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(10):2140 –2155, Oct. 2006.

[94] J. F. Ziegler. Terrestrial cosmic rays. *IBM Journal of Research and Development*, 40:19–39, January 1996.

[95] J. F. Ziegler and W. A. Lanford. Effect of cosmic rays on computer memories. *Science*, 206(4420):776–788, 1979.