

*Reliability-Aware Memory  
Design Using Advanced  
Reconfiguration Mechanisms*



Peyman Pouyan

Department of Electronics Engineering

Universitat Politècnica de Catalunya

A thesis submitted in partial fulfilment  
of the requirements for the degree of

*Doctor of Philosophy*

Advisor:

Prof. Antonio Rubio

Dr. Esteve Amat

---

Universitat Politècnica de Catalunya

September 2015







## Acknowledgements

I would like to thank my supervisor, Prof. Antonio Rubio, without whose valuable guidance and encouragement, this thesis work would have not been accomplished. From whom, I learnt how to make research and fulfill scientific collaboration. Beside that, I also learnt many precious life lessons such as always having a positive mind to problems and understanding that working intelligent is as important as working hard. I could never forget our conference trips to Hawaii, Mallorca, Grenoble, and Warsaw along with the thesis discussions.

I would then acknowledge my co-supervisor, Dr. Esteve Amat for his support and advises during all this work. Dr. Amat provided me many valuable suggestions and always helped me overcoming many technical obstacles, when we were writing papers together. His support was continuous throughout the thesis work, even accompanied this research work during his stay aboard, that was really admired to me.

In addition I would like to thank our research group members in UPC. Prof. Moll for his helpful comments on my presentations in our group meetings, and to Sergio Gomez, Nivard Aymerich, Carmina Almudever, Joan Mauricio, Jordi Perez, David Cavalleiro for their help and scientific discussions to fulfill this thesis work. Also I wish to thank our research collaborative members, Prof. Nafria and Dr. Martin from UAB for providing valuable advises.

Moreover, I want to thank my first electronic teacher, Prof. Ali Nasrabadi that motivated electronics as an interesting research

topic to me. Also I have to thank my friends Vahid Shafaei, Alireza Kharazian and Saeed Ghasemi for their accompanying and inspiring discussions during the years of my thesis.

Finally, I am deeply grateful to my father Nemat Pouyan, my mother Narges Pouyan, and my sisters Niloofar and Shaghayegh, for their continuous encouragement, love and support.

Dedicated:

To my first teacher of life, my mother Narges, for her  
unconditional love, support and encouragements during all these  
years.



## Abstract

Fast and complex data memory systems has become a necessity in modern computational units in today's integrated circuits. These memory systems are integrated in form of large embedded memory for data manipulation and storage. This goal has been achieved by the aggressive scaling of transistor dimensions to few nanometer (nm) sizes, though; such a progress comes with a drawback, making it critical to obtain high yields of the chips. Process variability, due to manufacturing imperfections, along with temporal aging, mainly induced by higher electric fields and temperature, are two of the more significant threats that can no longer be ignored in nano-scale embedded memory circuits, and can have high impact on their robustness.

Static Random Access Memory (SRAM) is one of the most used embedded memories; generally implemented with the smallest device dimensions and therefore its robustness can be highly important in nanometer domain design paradigm. Their reliable operation needs to be considered and achieved both in cell and also in architectural SRAM array design.

Recently, and with the approach to near/below 10nm design generations, novel non-FET devices such as Memristors are attracting high attention as a possible candidate to replace the conventional memory technologies. In spite of their favourable characteristics such as being low power and highly scalable, they also suffer with reliability challenges, such as process variability and endurance degradation, which needs to be mitigated at device and architectural level.

This thesis work tackles such problem of reliability concerns in memories by utilizing advanced reconfiguration techniques. In both SRAM arrays and Memristive crossbar memories novel reconfiguration strategies are considered and analyzed, which can extend the memory lifetime. These techniques include monitoring circuits to check the reliability status of the memory units, and architectural implementations in order to reconfigure the memory system to a more reliable configuration before a fail happens.







# Contents

<b>Contents</b>	<b>vii</b>
<b>Glossary</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xxv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 A Perspective of Device Scaling, Before, Now and Then . . . .	1
1.2 Embedded Memory Technologies . . . . .	4
1.3 Reliability in Digital Circuits . . . . .	6
1.3.1 Sources of Reliability Concerns in Nano-Scale Circuits .	6
1.3.2 Faults and Their Impact . . . . .	8
1.3.3 Approaches to Improve the Yield . . . . .	9
1.4 Thesis Motivation . . . . .	11
1.5 Thesis Objectives . . . . .	13
1.6 Thesis Organization . . . . .	14
<b>2 State of the Art of Modern and Future Technologies for Com- puter Memories</b>	<b>17</b>
2.1 Memories in Advanced Technology Nodes . . . . .	17
2.2 Modern CMOS and Emerging Memristive Devices . . . . .	18
2.2.1 FinFETs . . . . .	19
2.2.2 Memristors . . . . .	21
2.3 SRAM and Memristive Memory Architectures . . . . .	25

**RELIABILITY-AWARE MEMORY DESIGN USING  
ADVANCED RECONFIGURATION MECHANISMS**

---

2.3.1	SRAM Cell . . . . .	25
2.3.1.1	Standard SRAM Cell (6T SRAM) . . . . .	25
2.3.1.2	Other SRAM Topologies: 4T, 8T and 10T SRAM Cell . . . . .	27
2.3.1.3	Other Candidates for Fast RAMs: 3T1D . . . . .	29
2.3.2	SRAM Array Configuration . . . . .	31
2.3.3	Future Memristive Memory Architectures . . . . .	33
2.4	Faults in Memories . . . . .	36
2.5	Repairing Techniques in Memories . . . . .	37
2.5.1	Static Methods . . . . .	38
2.5.1.1	Error Correcting Code (ECC) . . . . .	39
2.5.1.2	Reactive Reconfiguration . . . . .	41
2.5.2	Dynamic Methods . . . . .	49
2.5.2.1	Periodic Cell Flipping . . . . .	49
2.5.2.2	Supply Voltage Scaling . . . . .	50
2.5.2.3	Adaptive Body Biasing . . . . .	51
2.5.2.4	Proactive Reconfiguration Strategy . . . . .	52
<b>3</b>	<b>Process Variability and Reliability Concerns in CMOS Tech- nologies</b>	<b>57</b>
3.1	Introduction . . . . .	57
3.2	Time-zero Variability . . . . .	58
3.2.1	Inter-die Variations . . . . .	58
3.2.2	Intra-die Variations . . . . .	59
3.3	Time Dependent Variability . . . . .	63
3.3.1	Bias Temperature Instability (BTI) . . . . .	63
3.3.2	Other Reliability Concerns in CMOS Technologies (HCI, TDDB) . . . . .	70
3.4	Time-zero Variability and BTI Aging Modeling in This Thesis	71
3.5	BTI and Process Variability in SRAM Memories . . . . .	74
3.6	Summary and Conclusions . . . . .	76

<b>4</b>	<b>Proactive Reconfiguration Concept in SRAM Arrays</b>	<b>79</b>
4.1	Introduction . . . . .	79
4.2	Analysis Framework . . . . .	80
4.3	Aging Evolution in a Non-Proactive Scenario . . . . .	81
4.4	Aging Evolution in the IBM'S Proactive Reconfiguration Tech- nique (Homogeneous Approach) . . . . .	83
4.5	Basis and Analysis of Variability-Aware Proactive Technique: A Static Non-Homogenous Approach . . . . .	85
4.6	Dynamically Adaptive Extension to the Non-Homogeneous Proactive Technique . . . . .	88
4.6.1	Overall Approach Flow . . . . .	88
4.6.2	Dynamic Recovery-Time Calculation . . . . .	90
4.6.3	Single Spare Column Proactive Reconfiguration Case . . . . .	91
4.6.4	Multi Spare Columns Proactive Reconfiguration Case . . . . .	95
4.6.5	Other Adaptive Proactive Approaches: Priority Sort, Based on Highest $V_T$ . . . . .	97
4.7	Impact of Adaptive Proactive Approach in SRAM Cell Design Parameters . . . . .	100
4.8	Architecture for the Proactive Technique and Overhead Eval- uation . . . . .	102
4.8.1	Architecture . . . . .	102
4.8.2	Control Unit . . . . .	107
4.8.3	Area Overhead Evaluation Results . . . . .	108
4.9	Summary and Conclusions . . . . .	110
 <b>5</b>	 <b>Monitoring Procedure in SRAM Arrays with Reconfigura- tion Mechanisms</b>	 <b>111</b>
5.1	Introduction . . . . .	111
5.2	Related Work . . . . .	113
5.2.1	IDDQ Testing . . . . .	113
5.2.2	NBTI on chip monitoring using BIST . . . . .	114
5.2.3	Few other approaches to monitor BTI aging in SRAM Array . . . . .	115

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

5.3	Proposed On-chip Variability and Aging-Aware Monitoring Technique . . . . .	117
5.4	Experimentation . . . . .	123
5.5	Summary and Conclusions . . . . .	129
<b>6</b>	<b>Process Variability and Reliability Concerns in Memristive Memories</b>	<b>131</b>
6.1	Introduction . . . . .	131
6.2	Memristor Models and Mechanisms . . . . .	132
6.2.1	Ionic Drift Model (HP model) . . . . .	132
6.2.2	The Pickett Model . . . . .	134
6.2.3	The Conductive Filamentary Switching Model (CF) . .	134
6.3	Reliability Concerns in Memristor Devices (ReRAM) . . . . .	136
6.3.1	Process Variability . . . . .	136
6.3.2	Endurance Failure Mechanism . . . . .	138
6.3.3	Random Telegraph Noise (RTN) . . . . .	140
6.4	Impact of Reliability Concerns in Memristive Memories . . . . .	142
6.4.1	Probability of Error in a Read Operation . . . . .	142
6.4.2	Probability of Error in a Write Operation . . . . .	146
6.5	Summary and Conclusions . . . . .	148
<b>7</b>	<b>Memristive Crossbar Lifetime Evaluation and Reconfigura- tion Strategies</b>	<b>151</b>
7.1	Introduction . . . . .	151
7.2	Memristive Lifetime Modeling . . . . .	152
7.3	Crossbar Lifetime Analysis . . . . .	154
7.3.1	Lifetime up to the First Failure . . . . .	155
7.3.2	Lifetime up to the First Failure Verified with Monte- Carlo Analysis . . . . .	156
7.3.3	Lifetime up to the Second Failure . . . . .	157
7.4	Non-Adaptive and Adaptive Reconfiguration in Memristive Crossbar . . . . .	160
7.4.1	Non-Adaptive Reconfiguration . . . . .	160

7.4.2	Adaptive Reconfiguration . . . . .	162
7.4.2.1	Determination of the Number of Shifts . . . . .	164
7.4.3	Comparison Between Non-adaptive and Adaptive Approaches . . . . .	169
7.5	Summary and Conclusions . . . . .	173
<b>8</b>	<b>Monitoring Approach in Memristive Memories with Reconfiguration Mechanisms</b>	<b>175</b>
8.1	Introduction . . . . .	175
8.2	Memristive Crossbar Architectures . . . . .	176
8.2.1	Write/Read Cycles . . . . .	177
8.2.1.1	Write '1' Cycle (SET) . . . . .	177
8.2.1.2	Write '0' Cycle (RESET) . . . . .	178
8.2.1.3	Read Cycle . . . . .	179
8.3	Monitoring of Crossbar Memristors . . . . .	180
8.3.1	Monitoring Scenarios in Crossbar . . . . .	181
8.3.1.1	Cell-by-Cell Monitoring . . . . .	181
8.3.1.2	Column-by-Column Monitoring . . . . .	183
8.3.1.3	Row-by-Row Monitoring . . . . .	184
8.3.1.4	Monitoring Approach Suggested in this Thesis	185
8.4	Non-adaptive and Adaptive Crossbar Reconfiguring Strategies	186
8.4.1	Non-adaptive Reconfiguring Strategy . . . . .	186
8.4.2	Adaptive Reconfiguring Strategy . . . . .	189
8.4.3	The Monitoring Circuitry Implementation . . . . .	192
8.4.3.1	Simulation . . . . .	194
8.5	Summary and Conclusions . . . . .	196
<b>9</b>	<b>Conclusion and Future work</b>	<b>199</b>
9.1	Summary of Contributions . . . . .	200
9.2	Future work . . . . .	202
9.3	Publications . . . . .	203
	<b>Bibliography</b>	<b>205</b>



# Glossary

**$V_T$**  Threshold Voltage.

**ABB** Adaptive Body Biasing.

**BISR** Built In Self Repair.

**BIST** Built In Self Test.

**BTI** Bias Temperature Instability.

**CDF** Cumulative Distribution Function.

**CF** Conductive Filamentary.

**CMOL** Complementary Metal Oxide Layer.

**CNFET** Carbon Nanotube Field-Effect Transistor.

**DEC** Double Error Correction.

**DFT** Design For Testability.

**DRAM** Dynamic Random Access Memory.

**DVS** Dynamic Voltage Scaling.

**ECC** Error Correction Codes.

**FBB** Forward Body Biasing.

## **RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS**

---

**FD-SOI** Fully Depleted Silicon On Insulator.

**FER** Fin Edge Roughness.

**FinFET** Fin Field Effect Transistor.

**FPGA** Field Programmable Gate Array.

**GER** Gate Edge Roughness.

**HCI** Hot Carrier Injection.

**HRS** High Resistance State.

**ITRS** International Technology Roadmap for Semiconductors.

**LER** Line Edge Roughness.

**LRS** Low Resistance State.

**LWR** Line Width Roughness.

**MGG** Metal Grain Granularity.

**MIM** Metal Insulator Metal.

**MLC** Multi Level Cell.

**MOSFET** Metal Oxide Semiconductor Field-Effect Transistor.

**NBTI** Negative Bias Temperature Instability.

**NW** Nanowire.

**OCAS** On Chip Aging Sensor.

**PBTI** Positive Bias Temperature Instability.

**PDF** Probability Density Function.



- PTM** Predictive Technology Model.
- RBL** Read Bit-Line.
- RCU** Reconfiguration Control Unit.
- RD** Reaction Diffusion.
- RDF** Random Dopant Fluctuations.
- ReRAM** Redox Resistive RAM.
- RTN** Random Telegraph Noise.
- RUF** Resource Usage Factor.
- RWL** Read Word-Line.
- SEC** Single Error Correction.
- SER** Soft Error Rate.
- SNM** Static Noise Margin.
- SOC** System On Chip.
- SOI** Silicon On Insulator.
- SRAM** Static Random Access Memory.
- STT-MRAM** Spin Transfer Torque Magnetostatic RAM.
- TDDDB** Time-Dependent Dielectric Breakdown.
- TDV** Time-Dependent Variation.
- TZV** Time Zero Variation.
- VCM** Valence Change Memory.
- VLSI** Very Large Scale Integration.



# List of Figures

1.1	Technological roadmap of scaling in recent time period and future . . . . .	4
1.2	Memory types and categories . . . . .	6
1.3	Bathtub curve . . . . .	9
1.4	Approaches to improve reliability inside memories [31] . . . . .	10
1.5	Structure of this thesis . . . . .	15
2.1	Schematic of a 3D FinFET Structure [35] . . . . .	20
2.2	a) Memristor structure, b) Example of a HP Memristive hysteresis behavior [40] . . . . .	22
2.3	Memristor types based on their voltage switching mode . . . . .	23
2.4	A Typical CMOS SRAM Cell (6T) . . . . .	26
2.5	<i>SNM</i> graphical representation [14] . . . . .	27
2.6	A 4T SRAM Cell . . . . .	28
2.7	A 8T SRAM Cell . . . . .	29
2.8	A 10T SRAM Cell . . . . .	29
2.9	A 3T1D Memory Cell . . . . .	30
2.10	SRAM Cells in array [14] . . . . .	31
2.11	SRAM Memory Block Diagram [43] . . . . .	32
2.12	1kB SRAM Memory Block Diagram [35] . . . . .	33
2.13	Example of a crossbar [47] . . . . .	34
2.14	Example of a memristive crossbar memory [50] . . . . .	35
2.15	Example of a memristive CMOL memory [51] . . . . .	35

**RELIABILITY-AWARE MEMORY DESIGN USING  
ADVANCED RECONFIGURATION MECHANISMS**

---

2.16 a) Hard faults by, defect density, opens, shorts, device break-down, [52], b) Transient faults by radiation in CMOS devices [27] . . . . .	37
2.17 Critical charge for SRAM/latch/logic [53] . . . . .	37
2.18 Reliability enhancement techniques in SRAM arrays . . . . .	38
2.19 Principle of ECC for a RAM [27] . . . . .	40
2.20 A memory array with flexible Intrasubarray replacement [27] .	45
2.21 A Memory array with concentrated spare line in intersubarray replacement technique [27] . . . . .	47
2.22 A block diagram of memory array repair with concentrated spare line in BISR memories [61] . . . . .	48
2.23 Periodic cell flipping . . . . .	50
2.24 Adaptive body biasing in SRAM array . . . . .	51
2.25 A block diagram of memory array for proactive reconfiguration [33] . . . . .	54
2.26 Lifetime extension versus area overhead of conventional techniques and the proactive reconfiguration [33] . . . . .	55
3.1 a) An example for $V_T$ variability at time zero, b) $V_T$ variability at different gate lengths [68] . . . . .	58
3.2 Random Dopant Fluctuation (RDF) in sub-micron CMOS technologies [70] . . . . .	60
3.3 LER in sub-micro CMOS technologies, [72] . . . . .	61
3.4 a) Fin Edge Roughness(FER), b) Gate Edge Roughness(GER), [18] . . . . .	61
3.5 Process variability in nano-scale CMOS devices [35] . . . . .	62
3.6 An example for $V_T$ variability shifting in time . . . . .	63
3.7 NBTI aging mechanism in PMOS transistors [75] . . . . .	64
3.8 $V_T$ shift due to NBTI and PBTI in PMOS and NMOS transistors [79] . . . . .	65
3.9 PMOS degradation and recovery behaviour [82] . . . . .	66
3.10 The recoverable (R) and permanent (P) damage components [83] . . . . .	67

## LIST OF FIGURES

---

3.11	Applying a positive bias to the PMOS transistor accelerates the degradation recovery [83] . . . . .	67
3.12	$V_T$ shift due to BTI under time and voltage acceleration in 14nm pFinFET [84] . . . . .	68
3.13	BTI recovery characteristics for PMOS NBTI in both 20nm planar and 14nm FinFET show similar recovery properties [85]	69
3.14	BTI recovery in FinFETs is more affected by stress time, devices with shorter stress time recovered faster [86] . . . . .	69
3.15	$V_T$ -shift evolution when samples are subjected to BTI stress. The stress time divisions are well defined . . . . .	73
3.16	$V_T$ -shift performance of a device subjected to BTI stress. The behaviour is divided into two phases: stress and recovery . . .	73
3.17	BTI wear-out recovery behavior with different recovery factors: (a) $Rf=0$ and (b) $Rf=30\%$ . . . . .	74
3.18	Measured butterfly curves of 1-kB SRAM cells [87] . . . . .	75
3.19	Combined effect of NBTI and PBTI on 6T SRAM $SNM$ in 45nm [79] . . . . .	76
3.20	BTI induced aging in Read $SNM$ for (a) FinFET and (b) MOSFET based 6T SRAM cells [23] . . . . .	76
4.1	An example of a SRAM configuration with five columns (four operational + one spare SP1) . . . . .	82
4.2	$V_T$ aging slopes of columns in a non-proactive approach in a SRAM configuration of five columns . . . . .	83
4.3	$V_T$ aging slopes of columns in a proactive (dashed lines) and a non-proactive approach (continuous lines). Almost 2X lifetime extension is obtained. . . . .	84
4.4	System behavior when an adaptive proactive technique is applied. The aging slopes change with adaptive recovery times, taking into account the process variability. Fresh $V_T$ -values are shown on the vertical axis. . . . .	86
4.5	Adaptive proactive approach flow . . . . .	89

**RELIABILITY-AWARE MEMORY DESIGN USING  
ADVANCED RECONFIGURATION MECHANISMS**

---

4.6 Adaptive proactive reconfiguration among 5 memory columns has resulted to uniform activity distribution and value convergence . . . . . 92

4.7 Lifetime improvement is a memory system with 4 operational columns and one spare, in different configurations of 1:Non-Proactive, 2:IBM-Proactive, 3:Adaptive-Proactive . . . . . 93

4.8 Lifetime improvement is a memory system with 8 operational columns and one spare, in different configurations of 1: Non-Proactive, 2:IBM-Proactive, 3:Adaptive-Proactive . . . . . 94

4.9 Lifetime improvement is a memory system with 16 operational columns and one spare, in different configurations of 1: Non-Proactive, 2:IBM-Proactive, 3:Adaptive-Proactive . . . . . 94

4.10 Multi-spare adaptive proactive reconfiguration . . . . . 96

4.11  $V_T$  aging slopes of columns in a adaptive proactive with multi-spare, 8 working and two proactive spare . . . . . 97

4.12 Priority sort proactive approach based on highest  $V_T$  . . . . . 98

4.13  $V_T$  aging slopes of columns in a priority sort proactive , the slope of aging of thershold voltages converges to a point along the lifetime . . . . . 99

4.14 Degradation behavior of weakest SRAM  $SNMs$  of memory columns due to BTI aging by utilizing adaptive proactive reconfiguration . . . . . 101

4.15 Architecture for implementation of the adaptive proactive reconfiguration technique in 1kB SRAM cache . . . . . 103

4.16 Structure of one SRAM column in the adaptive proactive reconfiguration . . . . . 105

4.17 SRAM cell connection to Word-lines . . . . . 106

4.18 The node voltages Waveform . . . . . 106

4.19 Reconfiguration control scheme . . . . . 107

5.1 Impact of time exponent change in Iddq degradation trend [102]113

5.2 IDDQ Monitor circuit for SRAM array [102] . . . . . 114

5.3 SRAM cell state for NBTI testability [103] . . . . . 115

## LIST OF FIGURES

---

5.4	SRAM cell aging sensor [105] . . . . .	116
5.5	The in-situ technique for in-field testing of NBTI degradation in SRAM cell [106] . . . . .	117
5.6	Process variation and aging sensing scheme in column 0, the inset shows the digital current source implemented by the current mirrors . . . . .	119
5.7	a) Monitoring simulation scheme, b) Transistors in their operating region . . . . .	121
5.8	Current decrease during $V_T$ shift, representing aging of a NMOS	122
5.9	Monte-Carlo simulation considering variability in access transistor . . . . .	123
5.10	Schematic view of the fabricated on-chip aging sensor . . . . .	124
5.11	a) Monitoring sequence of writing the value one in one of SRAM storage nodes and sensing the aging in the corresponding PMOS, b) Comparator output shift when changing body bias voltage of PMOS PL . . . . .	125
5.12	The fabricated chip floor plan . . . . .	126
5.13	The chip measurement setup in the lab, a) Schematic for Ic connections, b) The PCB connections in the lab . . . . .	126
5.14	Aging and variability measurement in a) PMOS 1 b) PMOS 2	129
6.1	Memristor switching mechanism based on ionic drift model, applying voltage across the memristor can change the size of doped region ( $w$ ) therefore changing the resistance state of the memritor . . . . .	133
6.2	Memristor switching mechanism based on conductive filamentary switching, the oxygen vacancies ( $Vo$ ) form a filament between the two electrodes, changing the resistance state of the device . . . . .	135
6.3	Monte-Carlo analysis of HP memristor model while considering variability effect in HRS and LRS values . . . . .	137
6.4	Endurance failure behavior in memristive devices [24] . . . . .	140
6.5	Current fluctuations in ReRAM because of RTN . . . . .	141

**RELIABILITY-AWARE MEMORY DESIGN USING  
ADVANCED RECONFIGURATION MECHANISMS**

---

6.6 a) HRS and LRS normal distributions and the reference point (Rth) sweeping along them, b) Pe according to the corresponding value of reference resistance . . . . . 143

6.7 a) HRS and LRS distribution move toward each other because of aging, b) Pe worsens by the aging . . . . . 144

6.8 a) The LRS (left) and HRS (right) probability distributions following a normal distribution b) Pe in function of different Rref values, the closer the two distributions the Pe getting worse 146

6.9 *Tw* according to variability introduced in LRS and HRS values, b) Pe graph according to the sweep of the Twref point . . 148

7.1 Endurance degradation behavior in HRS and LRS values shown together with process variation at origin. The point ( $\tau$ ) is where the ratio of HRS to LRS reaches the critical point in terms of reliability and is equal to K . . . . . 152

7.2 The PDF( $\tau$ ) for fresh memristors, the mean value for number of endurance cycles in this normal distribution of  $\tau$  is 1e6 and sigma is 80000 in terms of endurance cycles . . . . . 154

7.3 The  $\tau$  (individual fresh memristor lifetime) and g (crossbar lifetime up to first fail) probability distributions . . . . . 156

7.4 The g probability distribution verified with Monte-Carlo simulation . . . . . 157

7.5 The y (crossbar lifetime between first to second fail) probability distributions . . . . . 159

7.6 The y probability distribution verified with Monte-Carlo simulation . . . . . 159

7.7 Non-adaptive reconfiguring approach of a mxm=4x4 in NxN=16x16 memristive crossbar, where each shift is a complete mxm jump 161

7.8 Adaptive reconfiguring approach of an mxm=4x4 in NxN=16x16 memristive crossbar, where each shift is in respect to location of weakest unit . . . . . 163

7.9 Average number of shifts in analytic approach, for mxm=4x4, 8x8, 16x16 and NxN ranging from 8x8 to 160x160 . . . . . 165



## LIST OF FIGURES

---

7.10	Average number of shifts in Monte-Carlo, for $m \times m = 4 \times 4, 8 \times 8, 16 \times 16$ and $N \times N$ ranging from $8 \times 8$ to $160 \times 160$ . . . . .	166
7.11	Average number of shifts in semi-analytic, for $m \times m = 4 \times 4, 8 \times 8, 16 \times 16$ and $N \times N$ ranging from $8 \times 8$ to $160 \times 160$ . . . . .	168
7.12	Lifetime extension in adaptive reconfiguring approach versus non-adaptive with various standard deviation (std) values and different array configurations: a) $m \times m = 8 \times 8$ and $N \times N$ ranges from $16 \times 16$ to $80 \times 80$ is considered, b) $m \times m = 16 \times 16$ and $N \times N$ ranges from $32 \times 32$ to $160 \times 160$ is considered . . . . .	171
7.13	3D display of one row-block in $N \times N = 20 \times 20$ crossbar, when a) Fresh crossbar is affected by process variability, b) Lifetime of an aged crossbar when non-adaptive reconfiguration is used, c) Lifetime of an aged crossbar when adaptive reconfiguration is applied . . . . .	172
8.1	A circuit implementation with memristive memories in crossbar configuration . . . . .	177
8.2	Writing 1 inside the 1T1R cell . . . . .	178
8.3	Writing 0 inside the 1T1R cell . . . . .	178
8.4	Read mode based on current . . . . .	180
8.5	Monitoring in crossbar cell-by-cell approach . . . . .	182
8.6	Monitor in crossbar column-by-column procedure . . . . .	183
8.7	Monitor in crossbar row-by-row . . . . .	185
8.8	Non-adaptive reconfiguration architecture and implementation showing the monitoring units in the left and upper part . . . .	187
8.9	Addressing scheme inside the non-adaptive architecture for an example of 4 divided sub-sections of a global $n \times n$ crossbar . .	188
8.10	Monitoring flow in the non-adaptive . . . . .	189
8.11	Monitoring diagrams in the memristive crossbar . . . . .	189
8.12	An architectural implementation of adaptive reconfiguration in crossbar architecture, where an $m \times m$ structure shifts according to location of its weakest or near-failing unit inside $N \times N$ crossbar . . . . .	191

**RELIABILITY-AWARE MEMORY DESIGN USING  
ADVANCED RECONFIGURATION MECHANISMS**

---

8.13 Monitoring circuit used in non-adaptive and adaptive cross-  
bar architectures, it monitors the HRS and LRS values of the  
memristor . . . . . 193

8.14 Illustration of ReRAM model [141] . . . . . 195

8.15 Circuit simulation waveforms in HSPICE showing the current  
waveforms at each monitoring phase . . . . . 196

8.16 Monitoring circuit observations according to voltage divisions  
at the cell output . . . . . 196

9.1 A comparison of variation aware techniques in SRAM and  
memristive memories . . . . . 200

# List of Tables

2.1	Number of data bits and check bits in SEC . . . . .	41
4.1	Round robin dynamic ranges . . . . .	90
4.2	Example of a round robin reconfiguration procedure . . . . .	91
4.3	Memory lifetime extension in multi-spare adaptive reconfiguration in comparison with non-proactive . . . . .	97
4.4	Relative lifetime extension obtained by the three proactive techniques over the non-proactive technique for a memory system based on five columns (4 operative + 1 spare) . . . . .	99
4.5	Relative lifetime extension and $V_{min}$ reduction drift obtained by the proactive over the non-proactive technique for a memory based on different number of spare and operational columns	102
4.6	Modified Node Voltages at each stage operation . . . . .	107
4.7	Components Area Size Used in Monitoring Circuits . . . . .	109
4.8	Adaptive Proactive Monitoring Design Area in SRAM Array .	109
5.1	Write and sense signals to monitor the aging in PMOS 1 . . .	128
5.2	Write and sense signals to monitor the aging in PMOS 2 . . .	128
7.1	Average lifetime per shift in mxm structures . . . . .	169
8.1	Applied signals in the monitor and flip phase of the cell under monitoring . . . . .	183
8.2	Implementation Overhead of Adaptive Reconfiguring Approach	194



# 1

## Introduction

### 1.1 A Perspective of Device Scaling, Before, Now and Then

The design of electronic systems, based on solid-state devices started more than 50 years ago, has met different deep challenges during its roadmap. Among the different device technologies, the conventional planar bulk-CMOS technology has evolved and scaled down until the extremely miniaturized dimensions of tens of nanometer [1]. The exponential growth in the number of transistors on a die, following Moore's law [1], has been one of the main impulses of technology achievements during the last 50 years. The scaling trend allows having more devices on a single die, and therefore, the functionality per chip can be more complex, resulting also in higher performance microprocessors with larger cache capabilities. Moreover, it leads to smaller, cheaper and faster circuits that consume less power, [2] allowing the explosive development of many fields of application.

Talking about the MOSFET scaling, one major achievement has been to scale down the transistor channel length, considered as the technology node

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

[3]. The successful scaling down of the MOSFET was due to the excellent material and electrical properties of  $SiO_2$  such as its thermal, chemical stability and also large band gap [4]. However, starting from the 90nm technology node, it has been necessary to modify the transistor in order to improve mainly the mobility inside the device. For instance, this was done by implementing strain in the material inside the transistors by some approaches such as embedding SiGe in the PMOS Source/Drain, nitride stress-capping layer for NMOS and PMOS devices [5].

Thanks to strained silicon, technology scaled down successfully to 45nm, but then a new challenge emerged mainly due to high tunneling (leakage) current flowing through the thin gate insulator [4]. The gate leakage increased with the gate oxide scaling and higher source-to-drain leakage currents was generated because of the thermal diffusion of electrons [4]. In order to overcome this challenge, a new gate dielectric material with a higher constant dielectric constant (high-k) was added to device and resulted in significant gate leakage reduction in comparison with 65nm technology node [6].

As the channel length scaled below 32nm, a wide range of effects showed more relevance. These effects including high leakage, static power and significant reliability challenges such parametric variations due to manufacturing process has led to emergence of new competitive device structures to substitute classical bulk-CMOS such as Fully Depleted Silicon On Insulator (FD-SOI) [7] and FinFET [8].

FD-SOI is a planar technology, based on positioning an ultra thin layer of insulator (buried oxide) on the top of silicon base and constructing the transistor channel with an un-doped very thin silicon film [7]. It provides better transistor electrostatic characteristics versus bulk-CMOS and can efficiently reduce the device leakage currents.

The other current device alternative, FinFET, is a Fin-shaped Field-Effect Transistor, in which the gate structure warps around the channel, and therefore has more control over the channel current [9]. Now then, similar to the previous technology nodes, FD-SOI and FinFET will also face significant reliability challenges in the future of their roadmap scaling [1]. Therefore, new alternative devices and architectures are being explored to allow the

continuation of the scaling roadmap and technology progress. Some of these technology devices that can potentially scale down to few nanometer (nm) sizes and considered as future emerging devices include:

1. Tunnel FETs [10]. They are considered a promising switching device to replace conventional technologies mainly for ultra-low power applications, due to their improved electrical characteristics at low voltage levels (sub-0.25 V) [10].
2. Carbon nanotubes (CNFET) [11]. These devices are cylindrical carbon molecules that exhibit unique properties, making them potentially useful in various fields of nanoelectronic design. Their unique structure and formation can increase the durability of a nanoelectronic circuit in comparison with other materials [2].
3. Nanowires (NW) [12]. They are long thin wires manufactured with semiconducting materials, like silicon or germanium, and can be fabricated in few nanometer of diameter [2]. They can be utilized as interconnect wires or even as an active device.
4. Memristors [13]. It is a two terminal device, in which its resistance can switch between two non-volatile states of low and high and can be scaled down to nanometer (nm) size. It can be utilized in different applications such as: nonvolatile memristor memory system; digital and analog systems and neuromorphic system.
5. Molecular devices [2]. Finally the ultimate device for nanoelectronic design roadmap can be the molecular device. In this sense there exists some set of molecules, which can behave like diodes or programmable switches. One example of such molecules is the carbon based ones with unique electrical properties [2].

Regarding these, Figure 1.1 summarizes the scaling roadmap of nano-scale CMOS electronic devices since 2003 and the newly appearing current emerging devices.

# RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

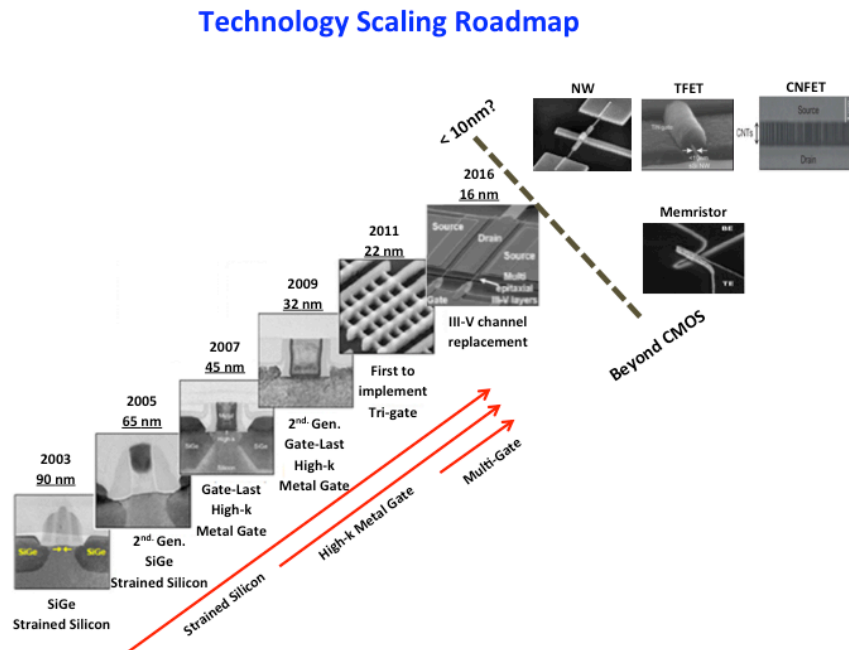


Figure 1.1: Technological roadmap of scaling in recent time period and future

In this context, one interesting fact in the scaling roadmap is that memory chips are usually the first to advance toward a new technology node in comparison with logic chips such as microprocessors, because they occupy a big area in the chip [3]. Therefore, memories have always been considered a critical circuit in the evolution of electronic design and studying their behaviour is of high interest for research community.

## 1.2 Embedded Memory Technologies

Embedded memories play a key role in modern microprocessors because of their impact in system performance and large silicon area occupation [14]. Among different kinds of embedded memories, Static Random Access Memory (SRAM) is one volatile memory type, which keeps the data without time limitation, and does not need periodic refresh.

SRAM cells are usually constructed with 6 transistors (often called as 6T memory cell), making them an expensive memory cell in comparison



with Dynamic Random Access Memories (DRAM), to be used in higher capacity applications such as inside the PC main memory [15]. Therefore, their main application is when high speed memory is needed such as inside the CPU caches and CPU register files [15]. Note that, they are usually built with minimal dimensions of technology node to increase the memory storage density, and this makes them a crucial component in terms of reliability inside the computers.

As the device dimensions scale to nanometer dimension nodes reliability aspects along with the increase of static power becomes the major challenges in SRAM chips. This makes it necessary to explore new memory technologies or configurations, which can possibly replace today's SRAMs, and could share their best characteristics.

Two possible candidates for future embedded memory technologies are the Spin Transfer Torque Magnetostatic RAM (STT-MRAM) and the Redox Resistive RAM (ReRAM) [1]. Both feature nonvolatile characteristic and can be potentially manufactured beyond 10nm technology node. STT-MRAM has better performance and endurance, on the other hand ReRAM structures are simple and almost utilize the CMOS fabrication process, very appealing in terms of cost and manufacturing [16]. Figure 1.2 depicts an overall view of current and future memory technologies [1].

Independently from the fact that which technology is used to manufacture memory chips, designing them in nano-scale sizes is highly affected by reliability parameters to be considered. Next section presents two of these reliability factors (variability and aging), their impact and the approaches to mitigate some part of their consequence.

# RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

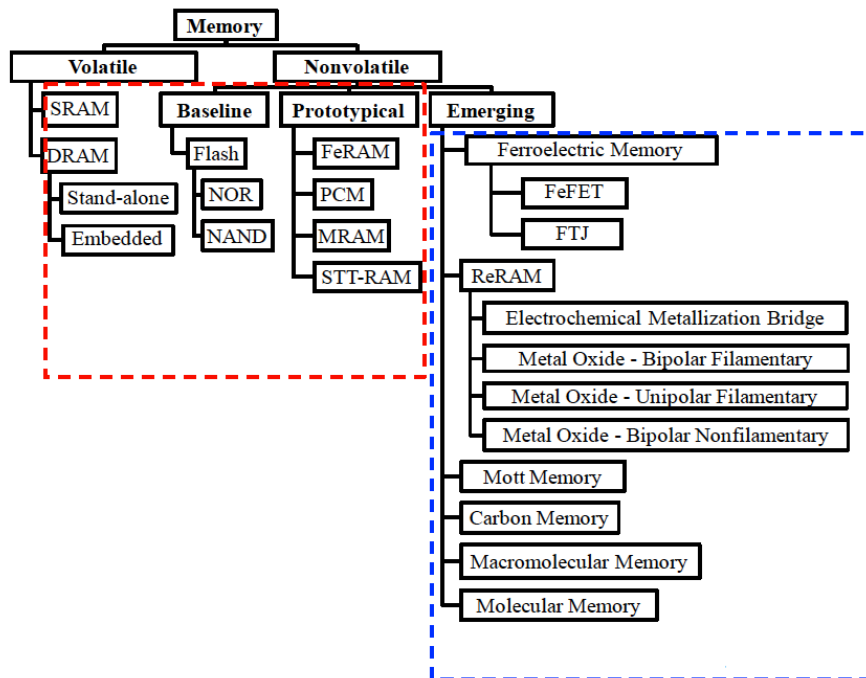


Figure 1.2: Memory types and categories

## 1.3 Reliability in Digital Circuits

While technology scaling has improved the performance of VLSI circuits, a variety of challenges have also emerged according to that, which have enhanced the necessity for design of novel circuits and innovative architectures.

### 1.3.1 Sources of Reliability Concerns in Nano-Scale Circuits

Process variation is consequence of device scaling [17], which has become more important below 90nm technology node. Nanoscale CMOS technology nodes are subject to a number of variation effects mainly caused by Random Dopant Fluctuation (RDF), Line-edge and Line-width Roughness (LER and LWR), variations in oxide thickness and others [17]. Note that although FinFET technology is less affected by process variation in comparison with bulk-CMOS technology, still some other factors of device manufacturing im-

pose variability in them (for instance variation in  $V_T$  values) [18].

As a consequence of variability, transistor parameters such as its threshold voltage ( $V_T$ ) will deviate randomly from its nominal value. It is worth to mention that, in the case of memristors, since resistive switching devices also use a CMOS-like manufacturing process, they are also affected by process variability. The source of variations in these devices includes different size filaments because of random electroforming process, and variations in electrode or oxide sizes. The key variation parameter in resistive switching devices would be its high and low resistance values.

Scaling also leads to an effective increase of stress factors, since an increase of power dissipation will result in higher temperature inside the chip and also bigger electric fields inside devices would impact failure mechanisms. One of these failure mechanisms is the aging phenomenon [17][19]. Three of the main aging mechanisms include Bias Temperature Instability (BTI) [20], Hot Carrier Injection (HCI) [21] and Time-Dependent Dielectric Breakdown (TDDB) [22].

BTI is an aging mechanism in which the transistor subjected to an operating bias, exhibits changes in its characteristics, such as its  $V_T$  and other electrical properties like drive current and transconductance are affected [20]. BTI has been significant at sub-90nm technology node of  $SiO_2$ -based PMOS transistors, named as Negative Bias Temperature Instability (NBTI). However, the use of high-k dielectrics to reduce gate leakage for 45nm technology node, has also caused the appearance of a degradation mechanism in NMOS devices, named as Positive Bias Temperature Instability (PBTI) [4]. In both NBTI and PBTI, the threshold voltage shifts during stress time, and unwanted threshold voltage shifts can cause failures. One important characteristic of BTI is that the device can experience some relaxation (reduction) from its  $V_T$  shift and some part of deviation can be recovered when the device is released from stress. Current advanced transistor devices such as FinFETs, also experience BTI aging, and it is reported that BTI aging in FinFETs is more aggressive than in the case of planar CMOS technology [23].

Similarly to advanced CMOS technologies, resistive switching devices are also prone to degradation, called endurance failure. The endurance degra-

## **RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS**

---

dation mechanism in resistive switching devices can be caused by too many defects such as oxygen vacancies accumulated during the cycling phase [24]. The endurance degradation phenomenon in resistive switching devices causes deviations in the resistance state of the device and could shift them toward each other [24]. This shift from nominal values in combination with process variability in resistive switching devices can impose reliability challenges and faults inside the electronic system [25].

Another reliability concern getting higher importance in nanoscale design is the effect of radiation on circuits [26]. Energetic particles such as alpha particles from packaging material and neutrons from the atmosphere may generate electron-hole pairs as they pass through the semiconductor device [26]. Then the according accumulated charge may flip the state of SRAM cell and cause transient fault inside the memory. This might become stronger due to further reduction of supply voltage and the device size, which would impose less stored charge inside the cell.

### **1.3.2 Faults and Their Impact**

With all sources of unreliability, such as process variation, circuit aging and others, two categories of faults can be defined, hard and transient faults [27]. Hard faults are a kind of faults that remain in the system permanently and the units that cause hard faults need to be replaced, otherwise the circuit cannot work properly or cannot operate at full performance. Hard faults may occur because of process variations [17], during manufacturing process or during the lifetime of the system, for example because of aging variations, which occur as a result of circuit being under stress [28].

On the other hand, transient faults due to radiation, power supply noise, etc. can cause temporary failures. Since system parts are not broken by transient faults, corrective operations are possible after rewriting the lost information. Furthermore, both process variability and aging mechanism can affect the system lifetime. There exist a graph called the bathtub curve [29], which represents a model for failure rate of electronic systems during their lifetime. Figure 1.3 presents an example of the bathtub curve where the first

region in the graph corresponds to the early failure defects that could occur during the early times of operation, because of for instance manufacturing defects. Then, there is the zone of working life in which the failure rate remains constant. Finally, the system lifetime reaches its ultimate zone in which failure rate increases due to aging effects inside the system. This wear out in the system along with some other degradation mechanisms can degrade their performance and cause failures.

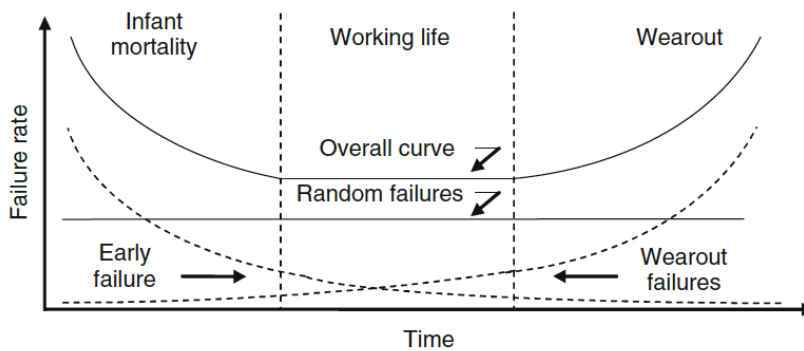


Figure 1.3: Bathtub curve

In this context, another important parameter is yield, which is defined as the ratio of the acceptable designs over the sample size and next section introduces approaches to enhance it.

### 1.3.3 Approaches to Improve the Yield

There are different approaches to increase the yield in electronic systems. These techniques include:

1. Information redundancy
2. Dynamic redundancy/Reconfiguration
3. Hardware redundancy

Information redundancy is a method based on providing extra information during the system functioning with some techniques such as parity and cyclic redundancy [30].

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

Afterwards, reconfiguration is a fault tolerance approach based on detecting the fault and its location and replacing the faulty unit with a redundant healthy unit. This technique would need a control system and redundant elements to replace the faulty units inside the system.

Finally, hardware redundancy is a technique in which the functional processing units are replicated and a voting circuit decides the correct output value based on redundant module outputs [30]. Among these techniques, the approach selection will depend on the expected yield and the device probability of failure.

For instance, Figure 1.4 presents an example for utilizing the appropriate approach in accordance with the device probability of failure ( $P_{PF}$ ) [31]. As the technology scales down the device probability failure increases and an approach with more overhead is necessary to fulfill the required reliability inside the memory system.

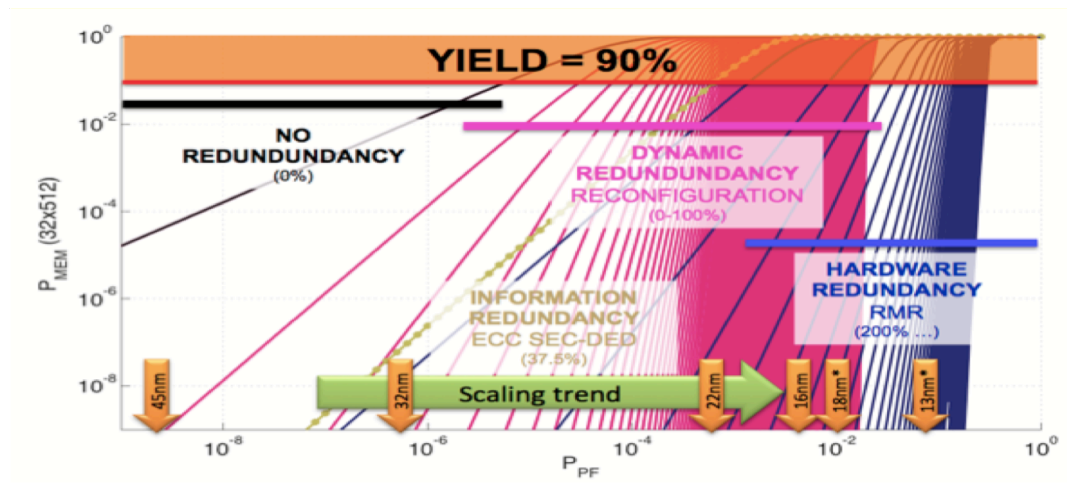


Figure 1.4: Approaches to improve reliability inside memories [31]

As shown in the graph above reconfiguration is an intermediate approach in terms of overhead to increase the yield in memories and improve their reliability. It's main drawback is that it cannot mitigate transient faults for example caused by radiation inside the system because these faults occur randomly in time and reconfiguration technique is not fast enough to handle them.

One example for a dynamically reconfigurable system is the TERAMAC computer built in HP labs [32]. This system was built with 864 Field Programmable Gate Array (FPGA) chips in which some of the FPGAs had some kind of defect, the system were able to test and reconfigure itself to work with high reliability and performance and therefore its overall functional lifetime was enhanced.

This thesis also aims to propose and design a dynamic reconfiguring approach at circuit and system level in order to improve the system reliability and extend the system lifetime.

### 1.4 Thesis Motivation

Integrated circuits an advanced nano-scale nodes, contain a large number of devices affected by reliability concerns such as process variation and aging. Therefore, new design techniques that cover these mechanisms and improve the system reliability are required, techniques, which are not necessarily covered by conventional design [27].

This thesis considers the analysis of two scenarios, the advanced CMOS technology, which is currently under production and will continue the Moore law for a certain number of future generations [1], and the Resistive Switching devices (ReRAM) considered by the ITRS as a potential alternative for "more than Moore" electronic systems in future.

In the field of design, among all units inside a System On Chip (SOC) and also full-custom chips, memory is the unit which occupies the largest area and is usually implemented with minimum technology node sizes. Therefore, it is a key issue to improve their reliability and extend their lifetime. This goal is the main aim of the present thesis.

Both SRAM and ReRAM memories are subjected to two reliability concerns: process variation and aging (BTI in SRAMs and Endurance degradation in ReRAM. BTI effect is a mechanism that can benefit from recovery properties, therefore developing approaches which allow the system elements to experience recovery mode, can extend their lifetime [33]. Unlike BTI aging, which can clearly benefit from recovery property, the recovery mechanism in

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

resistive switching memories devices is confined to special operative actions and limited. Therefore, some other novel approaches are required to improve their reliability.

Conventionally, one way to achieve the system reliability level is by designing system repairing mechanisms in a static approach [27], e.g. by having a fixed number of redundant elements. This reconfiguring approach is named as reactive reconfiguration. In such a reconfiguration mechanism a reconfiguring scheme allows the substitution of defective parts by the redundant ones, resulting in a fault-tolerant widely employed technique that is called reactive reconfiguration [33]. Although reactive reconfiguration could improve the yield very efficiently, as the number of unreliable components increases inside the system it is needed to move forward for building a resilient system, which can cope with stress and faults. Regarding this, an approach is required based on sensing the circuits by monitors and dynamically reconfiguring itself to avoid faults.

This thesis considers designing reliable nano-scale memory integrated circuits by using advanced reconfiguration approaches, in which the main goal is to enlarge the system lifetime in presence of process variation and aging effects. In relation with SRAM arrays the proposed reconfiguration approach in this thesis aims for enlarging the system lifetime by using the redundant elements (existing inside nowadays memories concentrated for yield enhancement) during the system lifetime in such a way that the system residual redundant elements take part in system operation during all lifetime up to system failure (proactive reconfiguration). This is a different approach than the reactive reconfiguration in which the redundancy is used only when replacing the failed units is required [33].

This thesis proposes designing aging detection circuits to monitor the aging status of memory elements adequate for the advanced reconfiguring mechanisms proposed. In order to have a reliable system that adapts itself to the vulnerabilities during its lifetime it is necessary to obtain information about status of operative elements. This could be possible by having an aging monitoring system. The monitored information will later be utilized in the reconfiguration mechanism and the overall cost in terms of area will



be evaluated.

In general and briefly, this thesis aims for the investigation of new design principles of adaptive reliability aware strategies, which utilizes reconfiguration to extend the embedded memory lifetime in accordance with faults, which might occur because of process variation and aging in both advanced CMOS and resistive switching technologies. Such dynamic and adaptive fault tolerance mechanism can utilize the existing resources efficiently in terms of allocated resources. This resource allocation is different in respect to the technology, for instance in SRAM memories it is based on dynamic recovery time management benefiting from BTI recovery properties, while in the ReRAM it is based on dynamic shifting and benefiting from high redundancy inside their structure.

### 1.5 Thesis Objectives

The main objectives of this thesis are as following:

- To design and investigate proactive reconfiguration algorithms and methodologies to extend the nano-scale SRAM memory lifetime, when it is subjected to process variation and BTI aging. The lifetime extension will be then compared between the respective algorithms to present and justify the optimized methodology based upon on the design costs and system requirements (Chapter 4).
- To evaluate the cost and efficiency of proactive reconfiguration implementation in memory, by analyzing the overhead in terms of area consumption. The evaluation is analyzed by hardware implementation of the proactive reconfiguration (Chapter 4).
- To propose new techniques and approaches of monitoring techniques to detect BTI aging at memory circuit level. These circuits will allow to utilize them to record the status awareness of memory units. The proactive reconfiguration mechanism uses the memory status awareness obtained from the monitoring parts for an appropriate dynamic

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

reconfiguration of the units (Chapter 5).

- To investigate the reliability challenge of nano-scale memory circuits for the emerging technologies and next generation of devices. These technologies include the resistive switching devices (memristor) and the aim is to analyze the impact of process variability and endurance degradation mechanism in the robust operation of a future memory cell based on these technologies (Chapter 6).
- To propose an approach to statistically estimate the lifetime of a single memristive device and a memristive crossbar in presence of process variability and endurance degradation. An analytic approach will be presented to model process variability and endurance degradation of a single memristive cell, then by utilizing a statistical approach the lifetime of the device will be predicted and verified by Monte-Carlo simulation (Chapter 7).
- To design and propose an advanced reconfiguration methodology in memristive crossbars in order to improve the memory reliability and extend the memory lifetime, which could be limited by endurance degradation and process variation. Regarding this, a monitoring circuit and approach is designed inside the memristive crossbars to monitor the memristive cell in respect to their variability and aging (Chapter 7-Chapter 8).

### 1.6 Thesis Organization

This thesis has been done in the framework shown in Figure 1.5, where it is possible to observe two parallel branches, one regarding the advanced CMOS technologies and another one regarding the resistive switching devices. Chapter 2 will present the state of the art for modern electronic devices and technologies. Then SRAM cell structures, SRAM array architectures, memristive devices and crossbar memristive memories are reviewed. Chapter 3 will analyze the main reliability concerns in advanced CMOS technologies

## CHAPTER 1. INTRODUCTION

---

such as process variability and aging, while its corresponding parallel Chapter 6 will analyze the same issues this time in resistive switching devices. Chapter 4 presents the adaptive proactive reconfiguration proposed in this thesis for SRAMs and its corresponding parallel chapter is Chapter 7 proposing reconfiguring approach in crossbar with resistive switching devices. In the following, Chapter 5 will present the monitoring approach in SRAM memories and Chapter 8 covers the same objective in resistive crossbars. Finally, Chapter 9 summarizes the work conclusions and results of this thesis, and also states the future work.

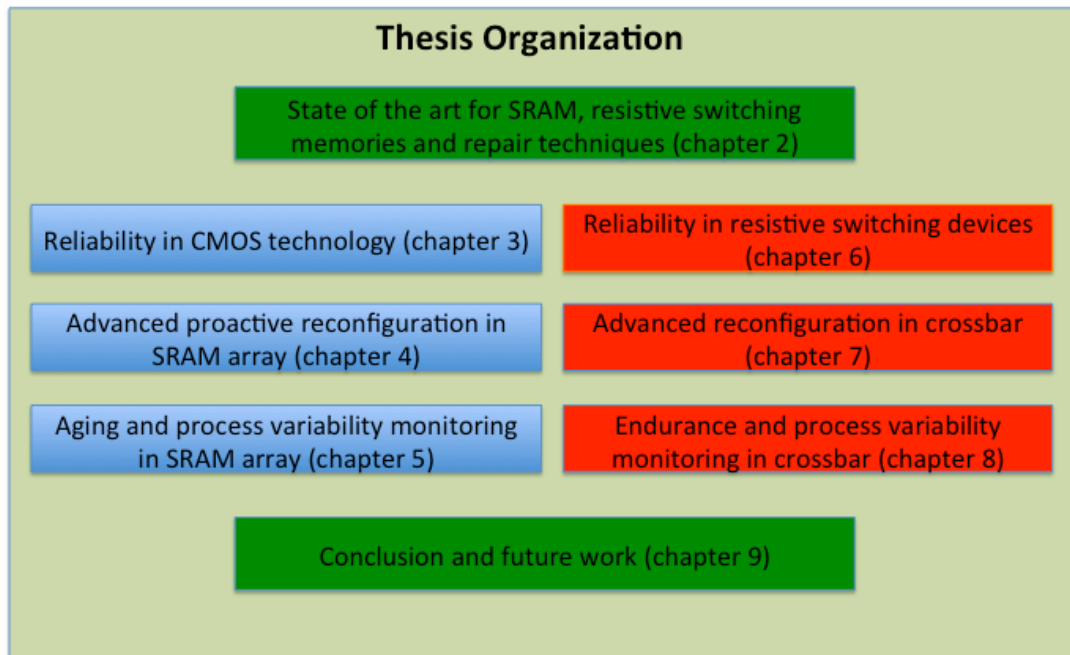


Figure 1.5: Structure of this thesis



# 2

## State of the Art of Modern and Future Technologies for Computer Memories

### 2.1 Memories in Advanced Technology Nodes

Today, semiconductor memories occupy a significant area in modern System on Chips (SOCs) circuits, and according to the International Technology Roadmap for Semiconductors ITRS [1], in near future, memories will occupy more than 90% of the cache-based microprocessors area.

Semiconductor memories are categorized into two groups: volatile and non-volatile. In the former, the stored data is lost when the power supply voltage is turned off. These memories are further divided into Dynamic RAM (DRAM) and Static RAM (SRAM). They are mainly used as main and primary storage, because of their higher speed in comparison with non-volatile memories. DRAMs have the characteristic of having low cost and high density, on the other hand SRAMs despite of higher cost feature higher speed [15].

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

Now then, non-volatile memories keep the stored value when the power-supply voltage is not available. Conventionally, they have been used as the secondary or permanent storage in the electronic devices, but with appearance of new devices their application might change in near future. ROMs, hard disks, flash memories and emerging technologies such as resistive switching devices are some examples of non-volatile memory [15].

### 2.2 Modern CMOS and Emerging Memristive Devices

As transistors channel length gets smaller toward nano-scale circuit design regime of sub 32nm, transistors are not able to act as ideal on-off switches, and instead, current leaks through them even when they are supposed to be off [34]. This excessive increase of sub-threshold and leakage current, results in higher power consumption and heat the chips up.

Generally, transistors have four terminals, source and drain, connected by a channel topped off by a gate and the bulk terminal underneath. In conventional CMOS design the biggest part of the transistor is in one plane, built into the silicon substrate. So then, when a gate voltage bigger than  $V_T$  is applied a conductive path in the channel is produced, allowing current to flow between the source and the drain. However, shrinking this structure further implies that removing the voltage on the gate no longer can completely stop this flow of electrons. As the distance of the source and drain decreases to only tens of nanometers (i.e.  $<20\text{nm}$ ) this short distance allows the electrons to leak through the lower part of the channel, farthest from the gate and cause wasted power and heat; and moreover it intensifies the short and charge channel effects [35].

This motivates the development of new structure for transistors as for example FinFET and other candidates to substitute the traditional bulk CMOS devices. Beside conventional approaches to develop memories in nano-scale design paradigm, researchers have always evaluated the possibility of utilization of other emerging devices with no leakage power, for memory

## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

---

applications.

One of such developing devices is the two terminal memristor, which can store data in its resistance state and have no leakage power. The emergence of memristive devices has pushed the technology to investigate new types of memory structures, which can be appropriate for their application. This section will present both FinFET and Memristor devices and will depict the memory architectures utilizing such devices.

### 2.2.1 FinFETs

Considering existing process variation and leakage currents in devices, such as the source-drain and the current between source/drain-bulk they will only get worse as chips shrink below 32nm. Therefore, researchers have proposed to plug the leak by raising the channel, source, and drain out of the substrate. The gate is then covering the channel on three sides like a lowercase "n". The current is now constricted only to the raised channel, and there is no path through which electrons can leak. Consequently, more control over the channel in FinFETs will lead to lower short channel effects and consequently lower device doping is required.

In this device the wrapped gate is like having three gates instead of one, therefore this type of new and modern transistor is called multigate. The most common multigate transistor design is a structure called a FinFET. In a FinFET, the channel connecting the source and drain is a thin, fin-like wall jutting out of the silicon substrate and Figure 2.1 presents a structure of a FinFET. Integrated circuit manufacturing by utilizing this device is currently under production (e.g. by Intel).

FinFETs are one of the emerging technology candidates to replace the planar transistors in the memories, as they are usually designed with the smallest channel size transistors in order to have the highest densities. It is worth to mention that FinFETs are more robust to parameter variability effects due to their lower doping. Moreover, they present better performance with lower voltages. These are other significant characteristics that make them attractive for future nano-scale sub 32nm design.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

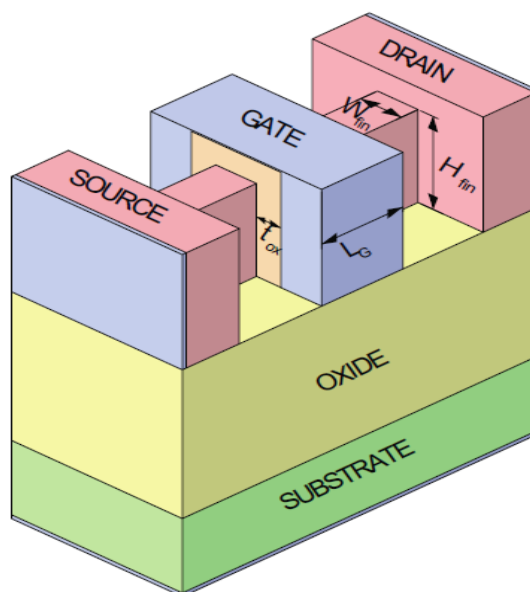


Figure 2.1: Schematic of a 3D FinFET Structure [35]

Note that, although FinFET technology is less affected by RDF, still some other sources of variability such as LER and Metal Grain Granularity (MGG) impose variability in them [36].

FinFET SRAM cells are built similarly to the planar SRAM cells and for instance the operation of a 6T SRAM cell is same as conventional planar bulk-based memory cell and the same design constrains need to be satisfied [36]. Moreover, memory architectures are also fully adapted to FinFET devices. However, the main difference between FinFET and planar SRAM cells is in sizing of the devices for robust operation where FinFETs cannot be sized similar to the conventional planar. This is because of fin-like structure of FinFETs, in which the width of the transistors can mainly be enlarged by increasing the fin numbers, or in some cases by increasing the quanta of the fin height [37]. Therefore, there have been research works that have considered this challenge and have proposed innovative techniques for design of robust FinFET SRAM cells [38].



## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

---

### 2.2.2 Memristors

A future promising device to make memories is the memristor, which is currently under prototyping [1]. The memristor's discovery goes back to about 40 years ago, when the properties of nonlinear circuit theory was being investigated by Professor Leon Chua at University of California Berkeley. Chua proposed the existence of a fourth element called the memory resistor by examining the relationships between charge and flux in resistors, capacitors, and inductors in a 1971 [39].

In accordance with other circuit elements such as Capacitor relating charge  $Q$  with voltage  $V(dq = Cdv)$ , Resistor relating voltage  $V$  with current  $I(dv = Rdi)$ , and Inductor relating flux  $\phi$  with current  $I(d\phi = Ldi)$ , the memristor relates the flux  $\phi$  with the charge  $q$  of the device ( $d\phi = Mdq$ ). His contribution also indicates that memristor presents a relationship between magnetic flux and charge similar to a resistor that gives between voltage and current.

Practically, memristor acts like a resistor whose value could vary according to the current passing through it, and which would remember that value even after the current disappears. Now then, the memristor did not receive enough attention by research community until recently, because the property of a material was too subtle to make use of it.

In 2008 HP presented a memristor constructed with titanium dioxide ( $TiO_2$ ) [40], which is a metal oxide, and it is highly resistive in its pure state. However, it can be doped with other elements to make it very conductive. In  $TiO_2$ , the dopants don't stay stationary in a high electric field; they tend to drift in the direction of the current. Putting a bias voltage across a thin film of  $TiO_2$  that has dopants only on one side causes them to move into the pure  $TiO_2$  on the other side and thus lowers the resistance. Running current in the other direction will then push the dopants back into place, increasing the  $TiO_2$ 's resistance. Figure 2.2a shows an structure of memristor device, moreover, the equations describing the memristive behavior in HP work was

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

in the form of Eq. 2.1 and Eq. 2.2:

$$V = R(w, i) \times i \quad (2.1)$$

$$\frac{dw}{dt} = f(w, i) \quad (2.2)$$

where  $w$  is a set of state variables,  $R$  and  $f$  can be explicit functions of time [40]. This model shows the hysteretic behaviour controlled by the intrinsic nonlinearity of memristance ( $M$ ) and the boundary conditions on the state variable  $w$ , depicted in Figure 2.2b.

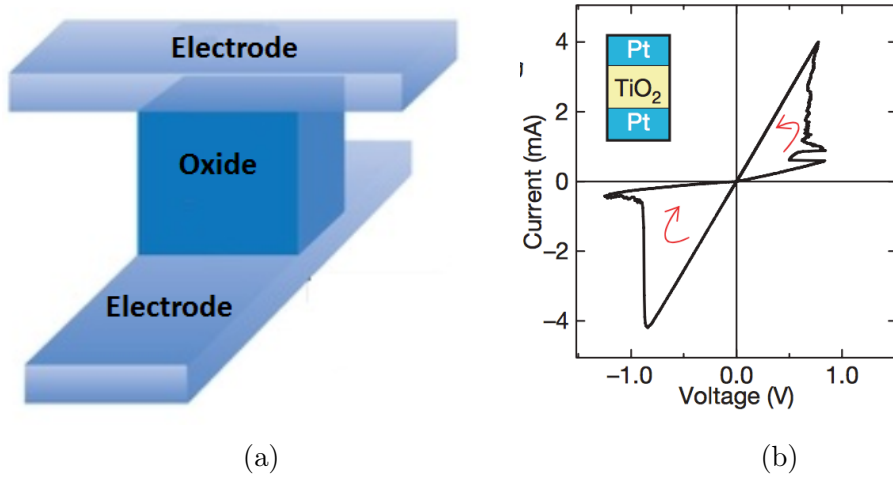


Figure 2.2: a) Memristor structure, b) Example of a HP Memristive hysteresis behavior [40]

Later in 2012, Chu defines memristor as any two terminal device that shows a hysteresis loop in the  $i - v$  plane by applying any bipolar periodic voltage or current waveform [41], in other words he said "If it is pinched it is a memristor".

Regarding this, some types of non-volatile emerging devices are categorized of being memristive devices such as: Resistive Random Access Memories (ReRAM), Polymetric memristor, Ferroelectric memristor, Manganite and Spintronic memristor [42]. They can all switch between low and high resistance non-volatile states, have low power consumption and high scalability,

## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

---

therefore they are very good candidates for future nano-scale memories.

However, these memristive devices have different characteristics in comparison with each other and in this context, reliability is one of the main design considerations in their application for memory systems. Among them, the resistive and spintronic memristors present promising specifications, and therefore has attracted many recent research works [42]. For instance, ReRAM devices are fast, low power, highly scalable and compatible with CMOS fabrication process [16].

Furthermore, according to ITRS, ReRAM devices are considered as an emerging memory device for the more than Moore era. Therefore due to interesting characteristics of ReRAMs, they have been chosen as the focus of this thesis for building reliable future memory systems.

### Redox Memory (ReRAM)

This type of memory device functions based on a change of resistance value on a Metal-Insulator-Metal (MIM) structure because of ion migration inside the structure along with some redox processes including electrode/insulator material [1].

In this sense, they usually need an initial electroforming process to form an elementary filament for future operation. Their switching speed is controlled by the ion transportation and they can be scaled down to few nm sizes. The ITRS categorizes ReRAM to four main types based on filamentary functioning and switching property (Figure 2.3) that is presented in brief in the following.

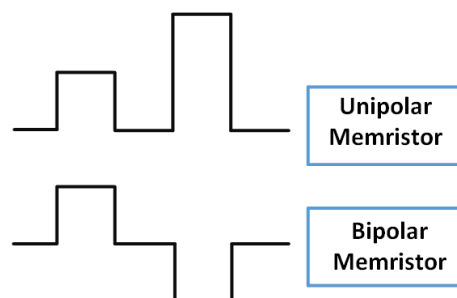


Figure 2.3: Memristor types based on their voltage switching mode

## **RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS**

---

### **Electrochemical Metallization Bridge ReRAM (CBRAM)**

The resistive switching operation in these devices is by utilization of electrochemical control of nano-scale quantities of metal in dielectric films; in order words a filament made by metal atoms is the basis of their memory function [1]. One CBRAM cell can be constructed by an electrode made of an active electrochemical material such as Ag or Cu, a passive electrochemical electrode such as Ta, Pt and a sandwiched electrolyte in the middle of two electrodes.

### **Metal Oxide-Bipolar Filamentary ReRAM, Valence Change Memory (VCM)**

This type of ReRAM is made by two electrodes and one insulator in the middle, where one electrode is to create the interface for switching and the other one is to create an ohmic contact and a place for storage of oxygen anions during the switching phase [1]. The resistive switching process in this type of ReRAMs is bipolar, meaning that a reverse voltage is needed to change the resistance states. This type of ReRAMs has attracted the most research among the other ReRAM types, and is the main focus in this thesis.

### **Metal Oxide-Unipolar Filamentary ReRAM**

Similarly to the previous ReRAM type, the switching process is by creation and disruption of a filament inside the device and between the two electrodes [1]. However, in this type of ReRAM the switching is unipolar, meaning that only one type of voltage but with different magnitudes is needed to change the resistance state of the device.

### **Metal Oxide-Bipolar Non-Filamentary ReRAM)**

In this type of ReRAM the switching process is not filamentary, and therefore the forming step is not needed [1]. A memory type functionality is observed for example by redistribution of oxygen vacancies because of electronic transport properties of the tunnel barrier. This type of ReRAM is less mature than the other types, and more research work is needed to analyze their switching properties and memory functionalities.

## **2.3 SRAM and Memristive Memory Architectures**

This section will present the structure for a SRAM cell and its alternatives along with their corresponding architecture. Next, in the following the architecture regarding the memristive memories is also presented.

### **2.3.1 SRAM Cell**

Embedded SRAM has a long reign in upper memory hierarchy than any other memories such as DRAMs because of its high random access speed performance [14]. Also with integration of multiple processing cores in one chip at nano-scale regime, the demand for integrated on-chip SRAM has been extended; therefore, SRAM arrays are more strategic for further analysis.

This section reviews in brief the state of art of SRAM block structure and a standard SRAM cell representative.

#### **2.3.1.1 Standard SRAM Cell (6T SRAM)**

One of the main advantages of SRAM cells is that they do not need to be periodically refreshed and can hold a 1 or 0 as long as they are powered in contrast to DRAM cells which need a more frequent data refresh. However, since SRAM cells contain a relative larger number of transistors, contacts and wiring, it takes more area than a DRAM on the chip (4X-5X) [14].

Figure 2.4 illustrates a typical 6T SRAM cell which consists of a pair of inverters (Q1 and Q3 inverter, Q2 and Q4 inverter) and 2 pass gate transistors (Q5 and Q6) used to access the cell during read and write operations. The pair of inverters is cross-coupled such that the output of one inverter is the input of the other one and vice versa.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

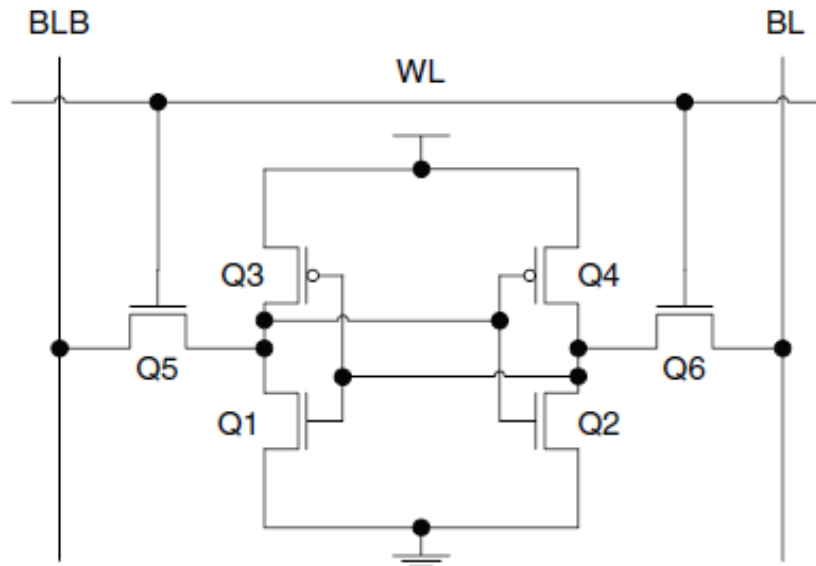


Figure 2.4: A Typical CMOS SRAM Cell (6T)

This SRAM cell structure is called a 6T SRAM cell since it has 6 transistors and it is one of the best memory cells in terms of robustness [14], low voltage and low power performances. Access to the cell is through the word-line signal and the two access transistors (Q5, Q6) that control the connection of the bitlines to the cell.

A SRAM cell is designed in such a way that ensures an expected long data hold (retention mode), a non-destructive read operation and a reliable write operation. More detailed description of the functionality and operation of the 6T SRAM cell can be found in [43]. This thesis considers the 6T SRAM cell as the baseline memory cell to work with in the following chapters.

One of the main stability metrics to consider in the design and analysis of SRAM cells is the Static Noise Margin (SNM). It is the maximum amount of noise that a cell can tolerate and still maintain the bi-stable operating points and retains the data (the data inside the cell does not flip) [14].

Figure 2.5 shows one of the approaches to graphically represent the SNM concept. The two voltage transfer characteristics of the SRAM inverters are inversed from each other and shape a butterfly curve. Then, in this

## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

figure, SNM is defined as the side of the largest square that can fit inside the butterfly curves, expressed in volts.

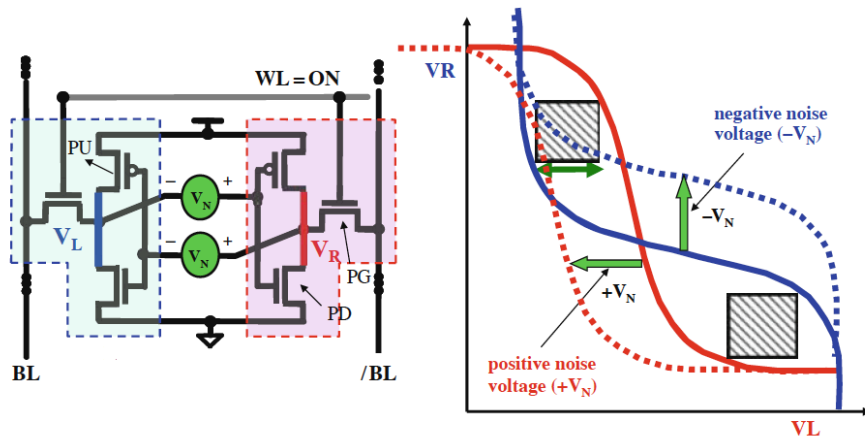


Figure 2.5: *SNM* graphical representation [14]

A cell is more robust if it has a larger SNM value, however due to the existing process variability and possible unbalanced aging in the memory cell devices, the symmetry of the cell and the SNM value can be easily deteriorated. Note that, there are two types of SNMs for data retention and read access named as Hold *SNM* and Read *SNM*. Now then, the 6T SRAM cell is more vulnerable to the noise during the read process, making the Read *SNM* more prone to be affected and therefore being considered in this thesis.

### 2.3.1.2 Other SRAM Topologies: 4T, 8T and 10T SRAM Cell

Beside 6T SRAM cells as the main type of SRAM in microprocessors other kinds of SRAM cells have also been introduced in microprocessors, either to reduce the size of standard cell or to increase the cell robustness in read or write operation. For instance, the 4T CMOS SRAM cell was proposed by NEC [44] for ultra-high density SRAM macros.

A key factor in design of 4T cells is that the leakage current of PMOS transistors should be much bigger than leakage current of NMOS transistors to fulfill the data retention requirement without a refresh. Figure 2.6 depicts this SRAM cell topology, where it benefits from utilization of minimal size NMOS and PMOS transistors.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

In a 4T SRAM cell the PMOS transistors serve as access transistors, which make the cell area efficient. A 4T SRAM cell occupies 50-65 % of a conventional 6T SRAM cell. However, the 4T SRAM cell is affected by reliability issues specifically in nano-scale processes with high process parameter variations [6].

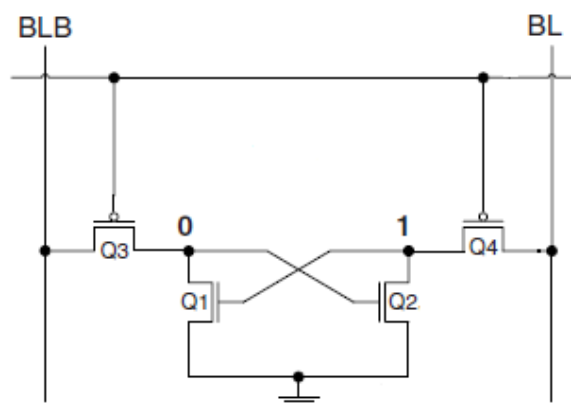


Figure 2.6: A 4T SRAM Cell

Another alternative to 6T cell is 8T SRAM cell which can improve the cell stability [14]. The 8T cell (Figure 2.7) is designed adding two NMOS transistors along with extra Read-Word-Line (RWL) and Read- Bit-Line (RBL) to the 6T cell. This cell can enhance the memory cell performance by improving their cell read stability, reducing the dynamic and leakage power, consequently, making it a good candidate for low power applications [14]. Adding the two NMOS transistors causes an increase in the memory cell area consumption, and also adds two new control lines to be managed by peripheral circuitry, which are drawbacks of using 8T SRAM cell.

Finally, 10T SRAM cell (Figure 2.8) is built by adding two more transistors to the 8T SRAM cell to improve SRAM cell access time [14]. It can give a much better design freedom than the 6T SRAM cell and reduce the power consumption, with an increase of cost due to higher area consumption.



**CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES**

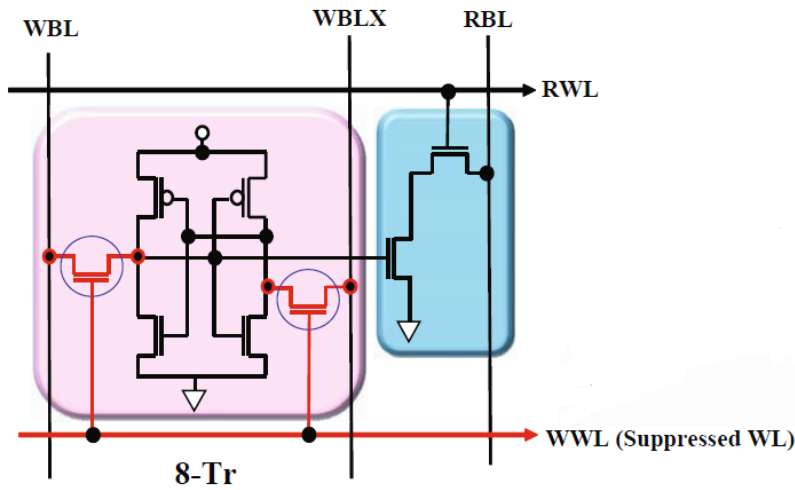


Figure 2.7: A 8T SRAM Cell

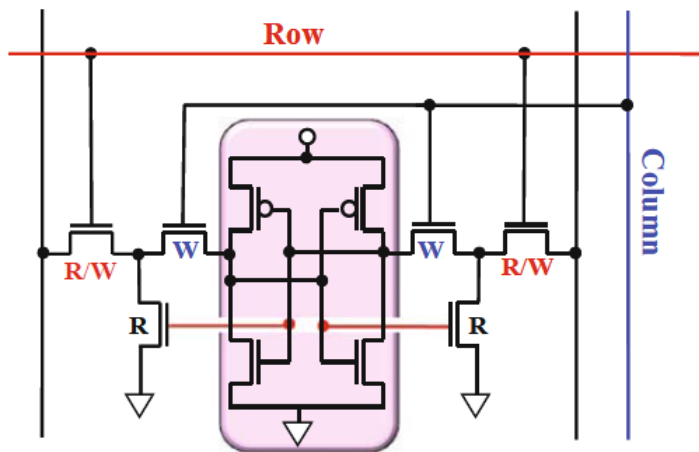


Figure 2.8: A 10T SRAM Cell

**2.3.1.3 Other Candidates for Fast RAMs: 3T1D**

Recently, multiple transistor DRAM cells, which use the capacitance of a transistor to store the memory state, are attracting attention to be utilized in embedded memories [45]. For instance 3T1D memory cell (3 transistors, 3T, and 1 one transistor in diode configuration 1D) is a memory cell constructed by only 4 transistors and has high performance read and write operation, which makes it comparable with the standard 6T SRAM cell. IBM proposed

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

this memory cell in 2005 and Figure 2.9 shows a schematic of this memory cell [45].

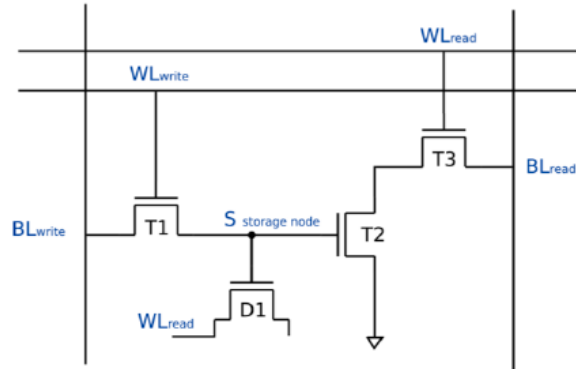


Figure 2.9: A 3T1D Memory Cell

In this cell, the bit is stored in the storage node S by the BL write line and T1 transistor. In contrast to other DRAM cells, the 3T1D has non-destructive read process because of existing a separate read path. It benefits from consuming less area in comparison with 6T SRAM cell (around 40% less) and presents to be more robust in existence of process variations [36]. These benefits make the 3T1D cell to be considered as one of candidates to replace SRAM cells in high-density cache memories [46].

## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

### 2.3.2 SRAM Array Configuration

So then, in order to utilize the SRAM cells in complex cache memory structures, they are organized in array configurations. In this context, Figure 2.10 presents an example of the SRAM cells arranged in an array of rows and columns. Each row of bit cells shares a common wordline (WL) and each column of bit cells shares a common bitline (BL). In a systematic view, the SRAM core is built by a number of arrays of  $M \times N$ , where  $M$  is the number of bits and  $N$  is the number of rows.

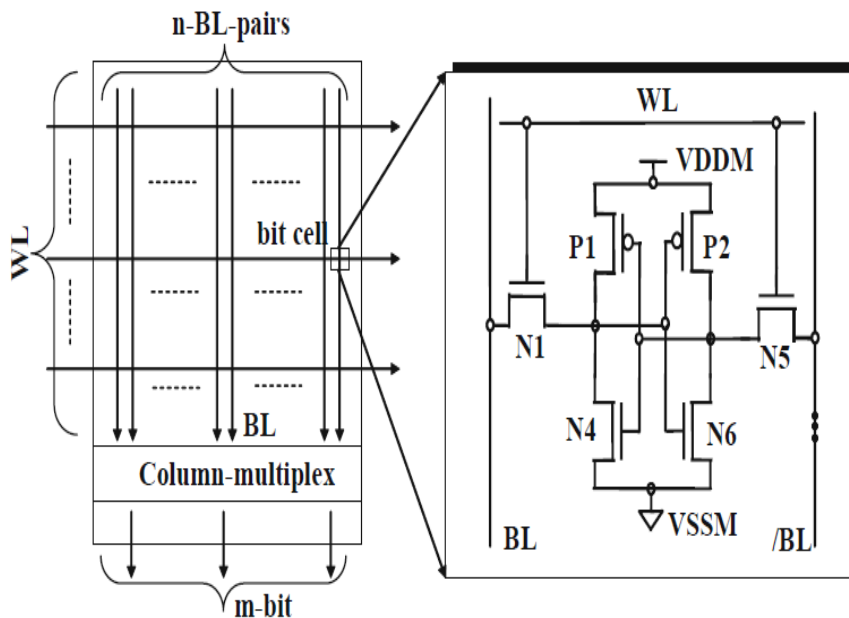


Figure 2.10: SRAM Cells in array [14]

Figure 2.11 depicts an example of a SRAM core where a row decoder decodes  $X$  row address bits and selects one of the wordlines, an additional  $Z$  decoder selects the accessed page, and column decoders addressed by  $Y$  address bits choose the appropriate bitlines.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

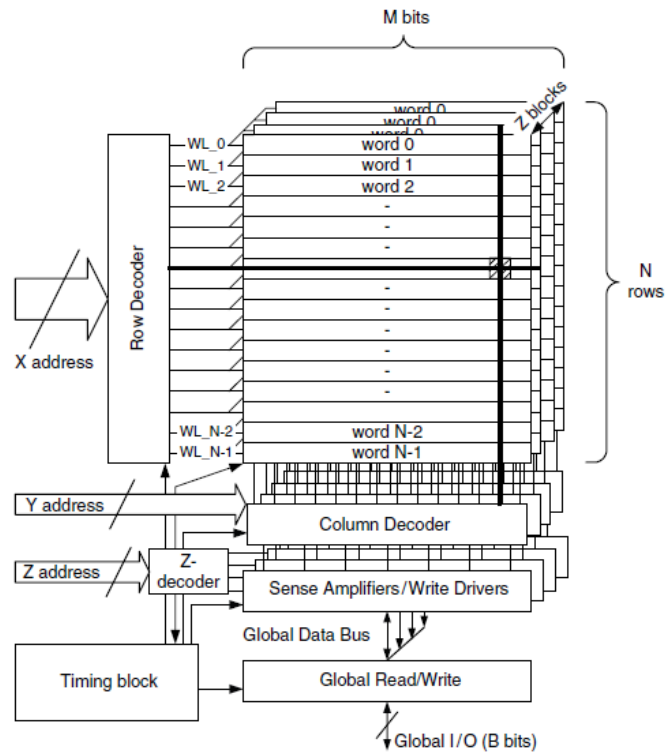


Figure 2.11: SRAM Memory Block Diagram [43]

This thesis has used the SRAM architecture proposed in [35] as an application example (the architecture shown in Figure 2.12). It is 1kB SRAM memory constructed from 6T SRAM cells, having 64 rows and 128 columns. The memory cells are addressed by the row decoder and column multiplexer and the sense amplifiers are utilized to read the status of the memory cells.

**CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES**

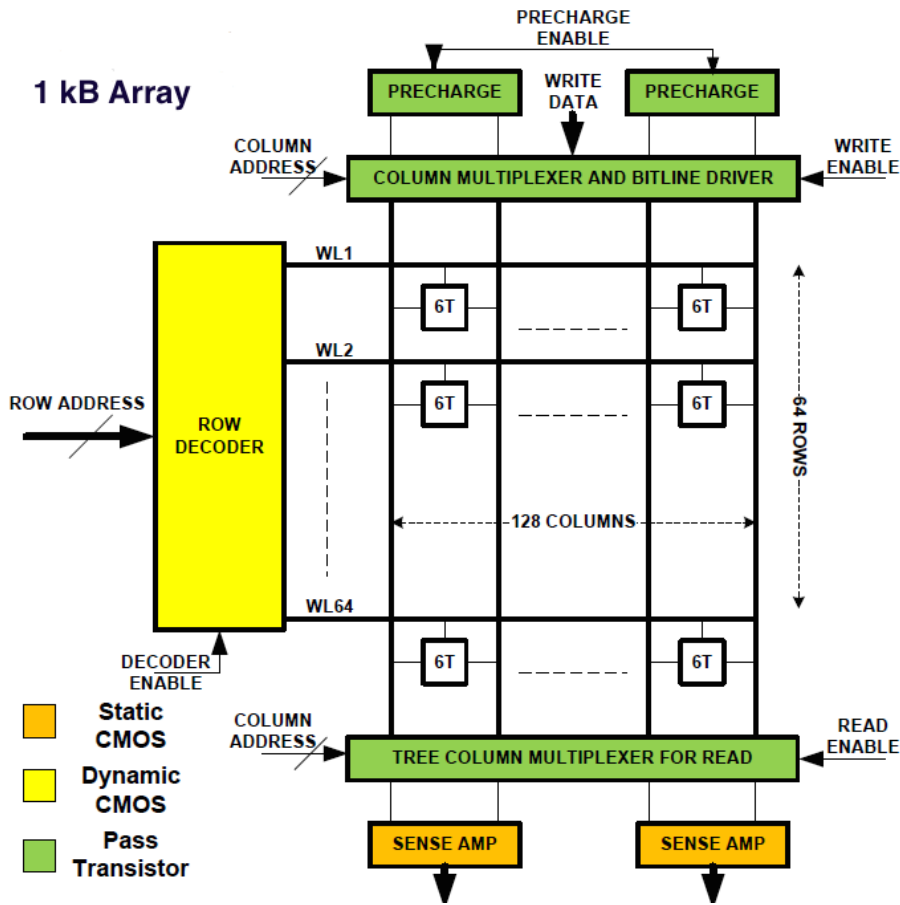


Figure 2.12: 1kB SRAM Memory Block Diagram [35]

On the other hand and with emergence of new devices as potential candidates for future memory configurations, novel structures are introduced in the literature, to build such memory architectures.

### 2.3.3 Future Memristive Memory Architectures

The simple and two-terminal structure of ReRAM allows fabrication of structures and architectures, which can offer higher density than SRAM ones [47]. One of the most promising nano-scale paradigms for future memory and computing architectures is the crossbar structure [48].

Nanowire crossbar memory is an architecture made of two layers of nanowires such as CNTs or SiNWs, and any intersection of two wires inside the cross-

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

bar can be configured by a two terminal switching device such as a ReRAM. Figure 2.13 shows an example of of a crossbar structure, which would allow increasing the memory densities up to hundreds of gigabytes by utilizing a uniform platform.

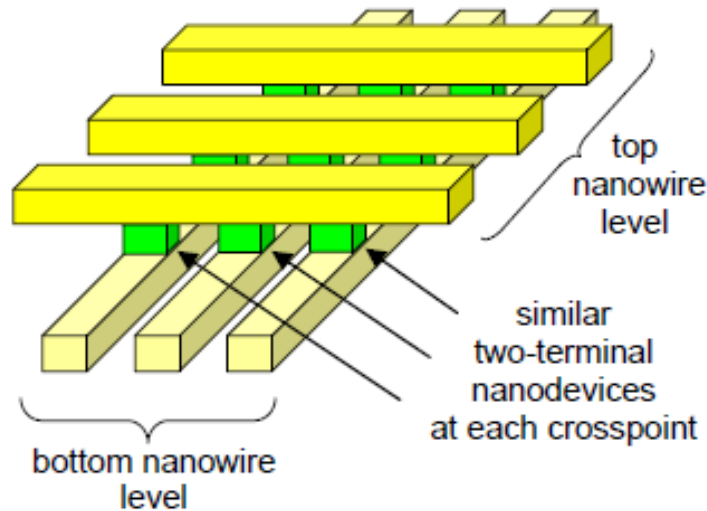


Figure 2.13: Example of a crossbar [47]

Nowadays, with the existing technology of memristive devices it is not possible to build a fully functional memory system and it would be needed to combine them with CMOS circuits in order to design memory array and the peripherals of the memory system [49].

Regarding this, Figure 2.14 shows an example of a hybrid memristive crossbar memory [50]. This thesis will be based on a hybrid crossbar memory, where the main array is constructed with memristive cells.

Another quite well known hybrid approach for making memristive crossbar memories is the CMOL design proposed by HP [51]. This approach utilizes crossbars on top of the CMOS cells, therefore combining the CMOS technology with resistive switching devices to build a highly dense memory system. In this structure, the nanowire and the two terminal devices are fabricated on top of the CMOS stack (Figure 2.15).

The challenging part of the CMOL architecture is the connection between

CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

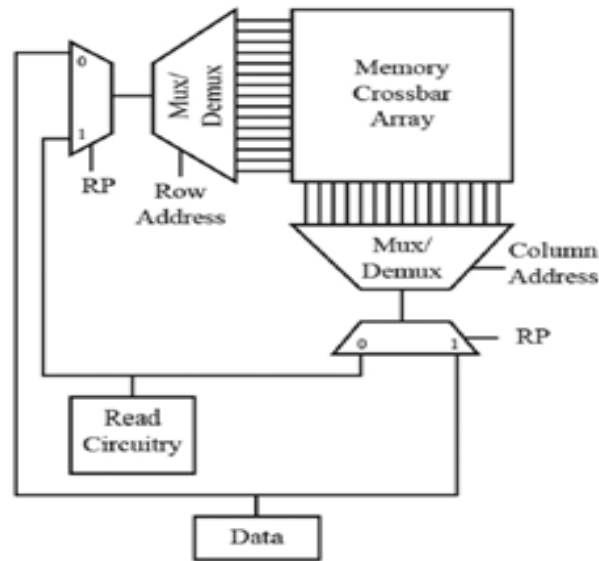


Figure 2.14: Example of a memristive crossbar memory [50]

CMOS and junction devices, as two sets of metal pins should penetrate into the nanowire to connect the top and bottom nanowires to the CMOS stack. Note that the CMOL architecture uses reconfiguration in conjunction with error correction approaches as the defect tolerance approach.

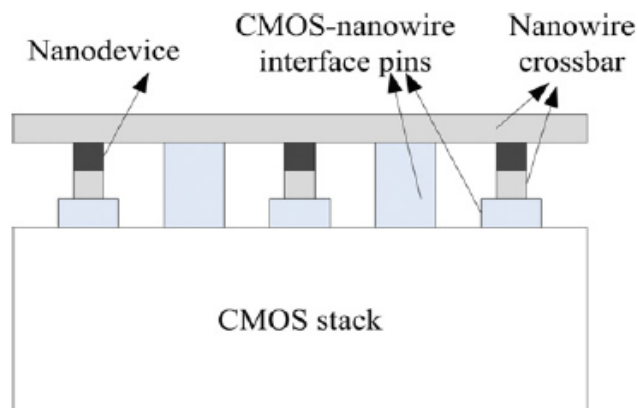


Figure 2.15: Example of a memristive CMOL memory [51]

## 2.4 Faults in Memories

Memories are built of electronic devices and scaling has caused the devices become more vulnerable and prone to faults. The memory faults are divided into two groups, hard and transient [27].

The hard faults make the affected unit unusable permanently and the unit needs to be replaced. Figure 2.16a shows an example of hard fault, a short circuit inside the chip. These faults are mainly caused by origin manufacturing, for example various fabrication-process defects or during the chip lifetime. With the technology getting smaller the rate of hard faults are increasing, because of getting more relevance in phenomena such as aging, occurring in the nano-scale regime. The hard faults in memories are mainly categorized to single-bit faults (when one memory cell is faulty), row or column faults (the cells inside the row or column are faulty), array faults (all memory cells in the array are faulty).

Unlike hard faults in which the unit gets unusable permanently; devices don't break in transient faults. When a transient fault occurs, the information in the node cell and other circuit nodes is lost, but it is possible to do correct operations after rewriting the lost information. For instance, transient faults could occur because of noise and incidents of alpha ray or cosmic ray. Figure 2.16b shows an example of transient faults happening by cosmic and alpha ray. If the charge of the ray is absorbed at the information node, an upset of the node voltage may occur.

Moreover, as the devices and the voltages in memories get smaller the required critical charge gets reduced causing an increase in soft error rate (SER). Note that, the problem of SER is more significant in SRAM arrays than DRAMs [53]. Regarding this, Figure 2.17 shows that the  $Q_{CRIT}$  of SRAM and logic circuits decreases with feature size.



## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

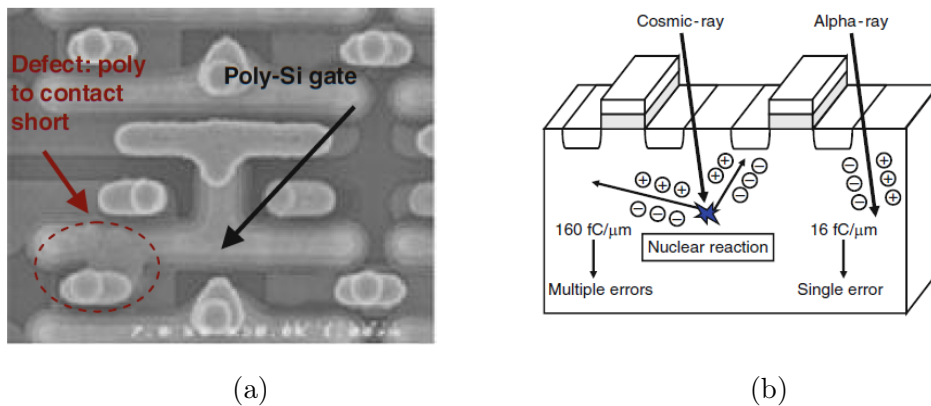


Figure 2.16: a) Hard faults by, defect density, opens, shorts, device breakdown, [52], b) Transient faults by radiation in CMOS devices [27]

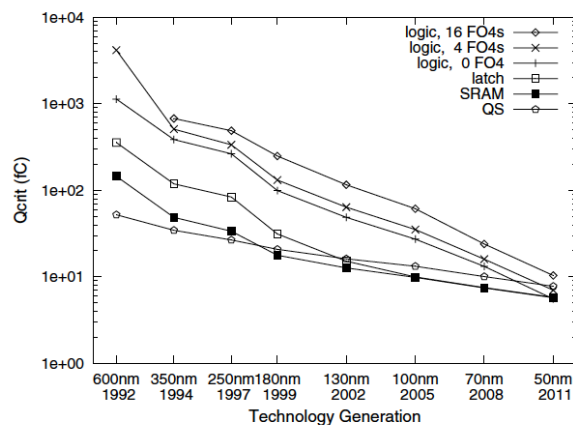


Figure 2.17: Critical charge for SRAM/latch/logic [53]

### 2.5 Repairing Techniques in Memories

The effective approaches to make SRAMs robust in front of variability and BTI aging in the literature [17] are categorized to two groups of static and dynamic methodologies. Static approaches, presented here, will then be followed by introducing the dynamic approaches. With scaling of device to nano-scale design paradigm and higher relevance of reliability concerns, it has been essential to also investigate dynamic approaches. These approaches will be presented in the following to enhance the reliability in SRAM memory

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

design.

Regarding this, Figure 2.18 shows some of the repairing techniques discussed here to mitigate the effects of process variation and BTI aging in SRAM memory cells. All these reliability concerns in nano-scale design paradigm, makes it essential to have repairing techniques in memories.

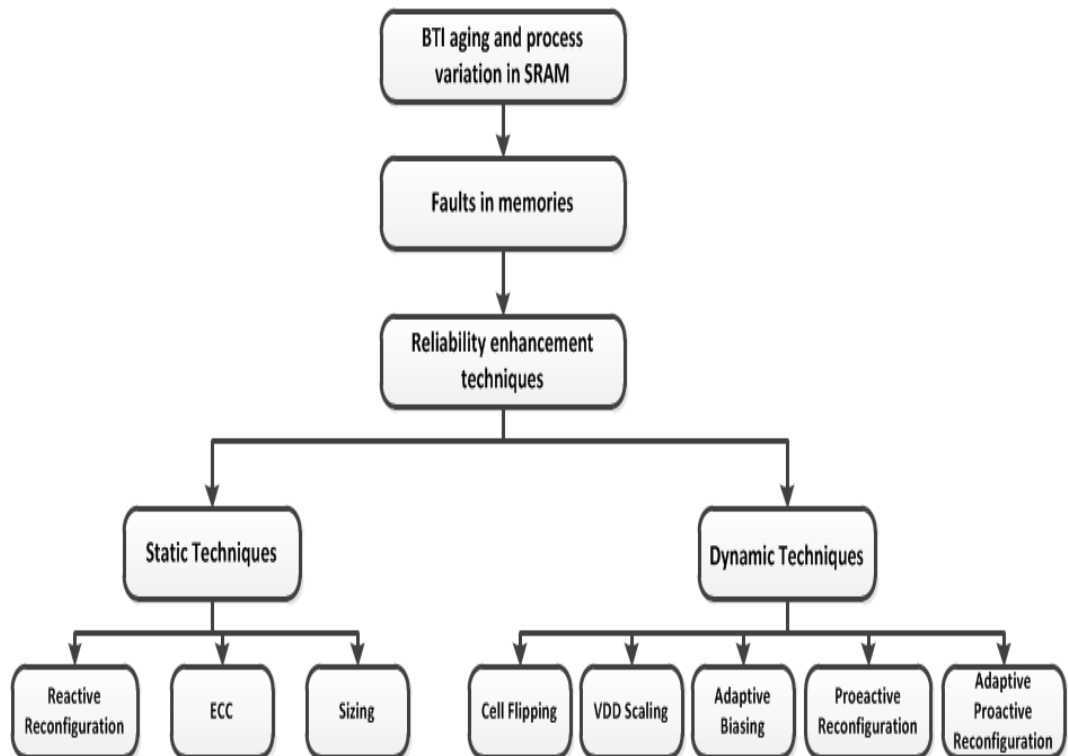


Figure 2.18: Reliability enhancement techniques in SRAM arrays

### 2.5.1 Static Methods

The static solutions are repairing techniques that are mainly considered at design time and can not adapt themselves to operating conditions. These approaches include such as Error Correcting Code (ECC), reactive reconfiguration, and transistor sizing.

The Error Correcting Code is a static technique to repair errors in the failed memories. It is based on information redundancy by containing extra memory cells in the chip used as check bits [54].

## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

---

Reactive reconfiguration is to use the redundancy available in the memory in a static way in which when an element fails the redundant spare units (rows, columns or arrays) will replace it.

The sizing approach proposes to design the length and width of SRAM transistors with some guardbands in order to make the cell robust in presence of process variation and BTI aging [55].

All these static approaches are considered as one time solutions at the origin design time and do not consider the device operation throughout its lifetime. In order to have reliable memory in the nano-scale design paradigm all the static solutions might lead to an increase in the area and power consumption. Therefore need for investigation and invention of efficient and strong reliable techniques for the design at nano-scale sizes has lead to emergence of dynamic reliability enhancement techniques.

### 2.5.1.1 Error Correcting Code (ECC)

Coding approaches are one of the most important and powerful architectural error protection mechanisms in recent computing systems [54]. Memories, as one significant unit of computing systems can also benefit from the Error Correcting Code (ECC) to become more reliable. Therefore, ECC is an effective technique utilized in memories to repair the transient and hard faults.

This approach is based on information redundancy by containing extra memory cells in the chip used as check bits. It encodes the input data during the write operation and generates some check bits. Then, the stored data in memory will have some information redundancy. If some faults occur in the memory before reading the data, for example because of alpha particles, the ECC can detect and correct the fault by the decoding circuits, if the number of faults does not exceed the correction capability of the specific utilized ECC approach.

The decoding operation works by first generating some check bits from the read data and comparing them with the read check bit. If they are both the same there is no fault in the data and if not the position of error is

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

detected and the faults are corrected before being outputted. Figure 2.19 shows the principle of ECC technique in memories.

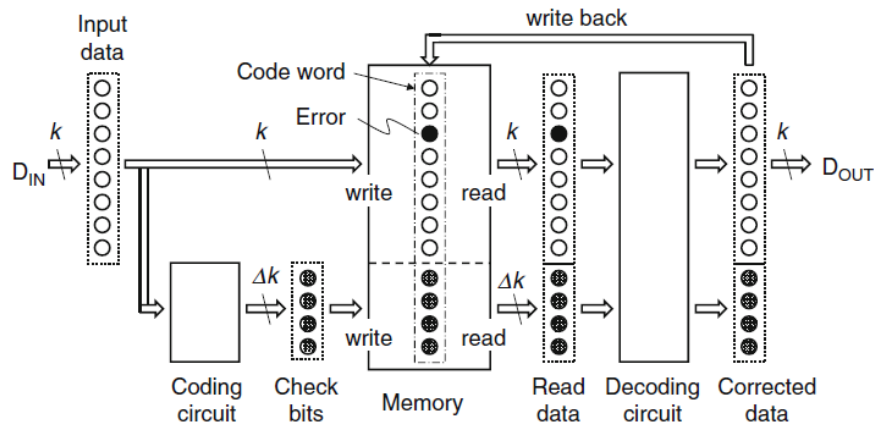


Figure 2.19: Principle of ECC for a RAM [27]

Generally, an ECC technique with higher error correcting ability has larger area consumption and access time penalty. The ECC approaches are categorized based on their error detection and correction capability. The three mostly used ECC approaches are the single error correction codes (SEC), single error correction and double error detection codes (SEC-DED), double error correction codes (DEC).

The bidirectional parity codes (in class of SEC) [56], the Hamming codes [57] and extended Hamming codes (in class of SEC-DEC) [58] are the ECC techniques that have been applied in RAMs until now. Previously, the DEC approaches were not efficient to be implemented in memories because of their large area consumption, however with scaling down the technology and increase in the fault rate they could be also an applicable approach for the repair of soft or hard errors in the memories.

The number of check bits ( $ckb$ ) for the ECC approaches reduces with increasing number of data bits ( $db$ ). However, the access time penalty increases with having bigger number of check bits. Therefore, there is a tradeoff between area and access time for using the ECC approaches in the memories. Table 2.1, as an example presents the number of required check bits for SEC approach.

## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

---

In comparison to the reactive reconfiguration approach, the ECC approach is more efficient in repair of random bit faults in the memories while the systematic faults such as row or column fault are more efficiently repaired by the reactive reconfiguration. Also ECC can repair both hard and soft errors but the reactive configuration can only repair the hard errors. Finally, the ECC technique uses larger area consumption and it has bigger the access time penalties in respect to the reactive reconfiguration.

Table 2.1: Number of data bits and check bits in SEC

Data Bits	4	8	16	32	64	128	256
Check Bits	3	4	5	6	7	8	9

### 2.5.1.2 Reactive Reconfiguration

Reconfiguration is a technique to improve the yield in memories [27] [59]. It is an effective method to reduce fabrication cost, but it also has some drawbacks such as increasing the chip size due to having some spare elements and reconfiguring routing and the control circuitry, and also reducing the performance by increasing the access time in memories. Consequently, there is always a tradeoff between improvements through using reconfiguration and the cost and penalties of having a reconfigurable memory.

To replace faulty elements with redundant elements in memory arrays to increase the yield was firstly discussed by Tammaru and Angell (redundancy for LSI yield enhancement) in 1969 [60]. They only considered the possibility of row redundancy, and just analyzed the randomly distributed defective rows then defective cells. One would immediately consider that the memory array yield would increase if both row and column redundancies are utilized. Therefore there are two main ways to add redundancy to the memories:

#### **Spare row or column:**

In this method the memory contains some spare rows or some spare columns, and when a fault is diagnosed the faulty row or column is replaced by the spare row or column [61]. Note that, spare rows are effective to repair faults

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

in a word line, and a word-line decoder, but spare columns are more effective for faults in a bit-line, column multiplexer, sense amplifier and column line decoding.

Therefore, the column reconfiguration has greater functional fault coverage. This approach is straight forward to implement and can operate in parallel with diagnosis algorithm but has low efficiency.

### **Spare row and column:**

One other approach to add redundancy is to add spare rows and spare columns to the memory simultaneously. This method typically needs a complete mapping of the memory before determining the replacement. When a fault is detected in the memory the faulty cell can be replaced by a spare column or by a spare row.

Although this approach is more efficient than the only row or only column approach, the complexity to find the optimal spare allocation is NP complete [62] and also a higher number of spare rows and columns is needed to achieve a sufficient chip yield. Considering the existence of redundant units in the memory array it is needed to analyze the replacement scenarios.

### **Replacement Scenarios:**

Globally, replacement scenarios [27] [63] for memories with array division are divided in three categories as:

- Intrasubarray
- Intersubarray
- Subarray

In the intrasubarray technique, the replacement of a failed unit with an operational spare one occurs in the same subarray and has benefit of having smaller access time penalty, while in the intersubarray scenario the replacement may be with a spare element in another subarray and benefits from better repair efficiency. The subarray replacement is a technique in which the whole subarray is replaced by a spare subarray.

A reconfiguration methodology usually has certain steps in which the whole reconfiguration process completes. This process first starts with know-

## **CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES**

---

ing the place of possible failing elements in the memory. It can be already programmed and recorded inside the memory by using nonvolatile memory or fuses or antifuses. Then, while accessing these elements it should recognize if they are faulty or not, and finally to replace the faulty element with the proper spare element and disable the faulty element. From circuit design perspective, reading the stored information about faulty or non-faulty elements and replacement scheme in memories is categorized in three groups:

1. Decoder programming
2. Address comparison
3. Shifting

Besides, there are two schemes to disable the faulty elements: I: Direct disabling and II: Indirect disabling, both they are only used with 1 and 2 replacement schemes. In overall there could be 4 possible replacement schemes:

1. Decoder programming and direct disabling
2. Decoder programming and indirect disabling
3. Address comparison and direct disabling
4. Address comparison and indirect disabling

In the decoder programming replacement scheme the address is programmable and spare decoders are utilized, meanwhile in the address comparison replacement scheme the spare elements are selected by using comparators. More details regarding the replacement structures can be found in [27].

In the following the Intrasubarray and Intersubarray replacement scenarios are presented. Then a static reconfiguration approach based on Intersubarray replacement will show another way of utilizing spare parts for memory repair.

**Intrasubarray replacement:**

## **RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS**

---

First, with intrasubarray replacement methodology there is needed to define two terms, replacement unit and replacement region.

Replacement unit: A set of memory cells replaced simultaneously. This unit can be a memory row or column or a set of memory rows and columns.

Replacement region: The area in which the memory row or column replacement occurs. In an intrasubarray replacement the replacement region is a subarray and it is couple of subarrays in intersubarray replacement.

The intrasubarray replacement method can be utilized in three different approaches, each one having their benefits and defects.

1. Simultaneous, where the number of address comparators is equal to the number of spares. The faulty word addresses are programmed in the address comparators and compared to the input address.
2. Individual, where every spare line in every subarray has its own address comparator and the replacement unit is one wordline. In comparison to simultaneous approach this technique utilizes bigger number of address comparators but on the other hand it benefits from lower probability of a fault on a spare and also requires less spare word lines if faults are randomly distributed. The efficiency problem in the simultaneous and individual replacement can get serious when the memory density increases.
3. Flexible, which is presented in Figure 2.20, solves some of the problems of simultaneous and individual replacement scheme. In this approach the replacement unit is one wordline since each address comparator compares both intra and inter subarray address bits. In this example, the faulty wordlines of  $W_0$ ,  $W_1$ ,  $W_2$ ,  $W_3$  are replaced by spare wordlines of  $SW_{00}$ ,  $SW_{01}$ ,  $SW_{11}$ ,  $SW_{20}$ , respectively.

In contrast to the simultaneous and individual replacement techniques that have fixed relationship between number of spare wordlines and address comparators, the flexible replacement technique benefits from a flexible relationship. This results in good usage efficiency in both spare line and address comparators and also higher probability of fault repair.



**CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES**

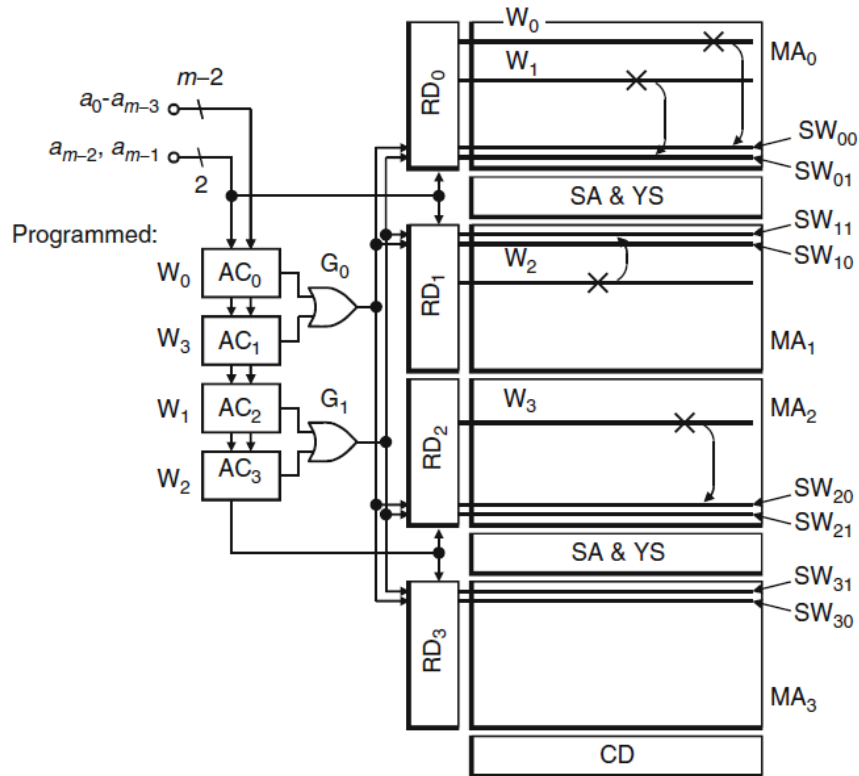


Figure 2.20: A memory array with flexible Intrasubarray replacement [27]

**Intersubarray replacement:**

The probability of cluster faults increase in high-density memories with existence of many memory array divisions. The number of repairs in a subarray is equal to the number of spare lines in the subarray. Therefore, to cover all cluster faults the number of spare lines could increase dramatically which results in chip-area cost.

This motivates an approach in which it is possible to replace the faulty line with a spare line in any subarray, inside the entire memory chip; such a replacement technique is called Intersubarray replacement. The Intersubarray replacement benefits from a higher and more flexible repair rate in the memories in comparison with intrasubarray technique with the drawback of higher access time penalty. This is because of the necessity to changing the activate subarray, due to change in the result of address comparator. The

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

Intersubarray replacement is divided to two classifications: 1-The distributed spare line 2-Concentrated spare line

In the first approach, every faulty line in each subarray can be replaced by the available spare lines in any of the subarrays. The number of possible repairs is equal to  $L \times M$ , which is equal to the number of address comparators  $R$ , where  $L$  is the number of spare lines and  $M$  is the number of subarrays.

In the Figure 2.21 depicted here, five clustered faulty word lines  $W_0$ - $W_4$  are replaced by the spare wordlines in subarrays  $MA_0$ ,  $MA_1$  and  $MA_2$ . For a successful repair that would be sufficient to have spare lines  $L$  equal to average number of faulty lines in a subarray.

In this approach all spare lines are located in one specific subarray and each subarray does not have spare lines in itself. For instance, in this example  $MAs$  contains the spare wordlines. A faulty spare line in each one of the subarrays can be replaced by one of the spare lines in the  $MAs$  subarray. The number of address comparators is equal to  $L$  as well as the number of possible repairs. This technique benefits from more flexible selection of spare lines and is more efficient in the number of address comparators. The drawback of the technique is the need for additional circuitry for the spare line subarray  $MAs$ .

**CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES**

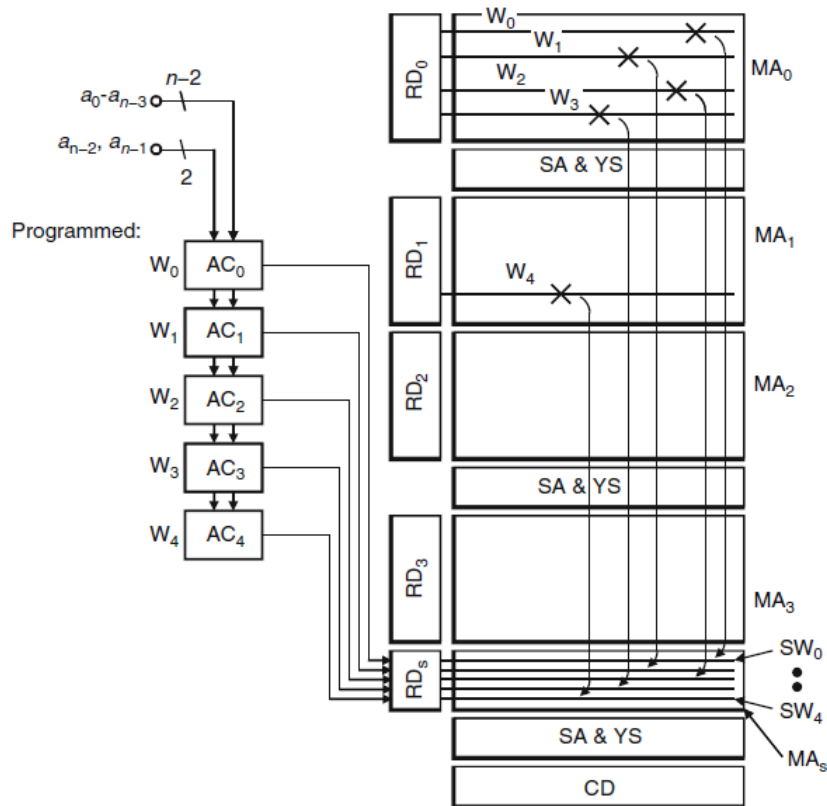


Figure 2.21: A Memory array with concentrated spare line in intersubarray replacement technique [27]

One other example for intersubarray concentrated memory array repair is its utilization in Built in Self-Repair technique. Built in Self-Repair (BISR) is an approach that extends the Built in Self-Test (BIST). BIST is a special design for testability technique that facilitates internal test pattern generation and output response compaction [43]. Regarding this, BISR will diagnose and repair the faulty units with replacing them with redundant elements in the high-density SRAM memories [61]. The reconfiguration elements of such an approach are:

1. A standard SRAM array of memory cells
2. Redundant memory columns

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

3. A BISR control unit (BBCU)
4. A memory reconfiguration control unit (RCU)

Figure 2.22 exhibits one example of architecture for a basic memory repair.

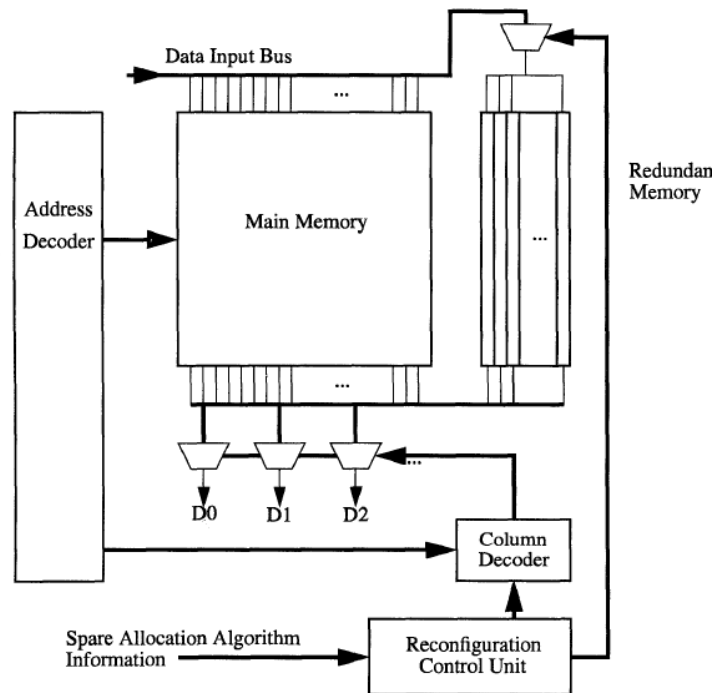


Figure 2.22: A block diagram of memory array repair with concentrated spare line in BISR memories [61]

The BBCU (not shown in the figure) has finite state machines that control the whole repair operation. The BBCU first tests all memory cells and stores the address of these faulty cells in the RCU. The RCU is a memory that its output controls the signals of multiplexers. So, when a faulty column is diagnosed it will be replaced by one of the spare columns. A single fault is repaired in per test phase. This is done by input multiplexers, which route the data to the correct functional columns and output multiplexers to output the data from operational columns.

## **2.5.2 Dynamic Methods**

The dynamic solutions have the benefit over the static ones as they can be adapted to the status of the SRAM cell during its lifetime. Therefore, they can save area and power overhead. Some of these approaches include the periodic cell flipping technique, standby supply voltage scaling, adaptive body biasing and proactive reconfiguration.

### **2.5.2.1 Periodic Cell Flipping**

SRAM cells may store a bit for a long time, and this could result in aging mechanism in one pair of the transistors that experience stress voltage. In [64] the flipping technique proposes to flip the bits stored in each SRAM cell between the two cross coupled inverters in order to mitigate part of NBTI aging.

This approach was originally is based on balancing the signal probability in the SRAM cells by a regular switch in the PMOS storage node; and could cause some recovery of the threshold voltage shift. The cell flipping technique can be implemented both in software and hardware based on the design necessity perspectives. The software implementation benefits from no cost in area, but the memory access time increases respectively. The hardware implementation is much faster in comparison with software implementation, but it needs some added circuitry and therefore has area overhead.

As the technology scales down toward sub 45nm sizes, utilization of high-k dielectric material in devices has turned PBTI to be an important reliability concern, and the aging of NMOS transistors have to be taken into account. However, in the cell flipping method always one pair of PMOS and NMOS transistors is under stress which impacts the cell aging in overall. Figure 2.23 illustrates the pair of transistors in the SRAM inverters that are under stress every time the data is exchanged data between the pairs.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

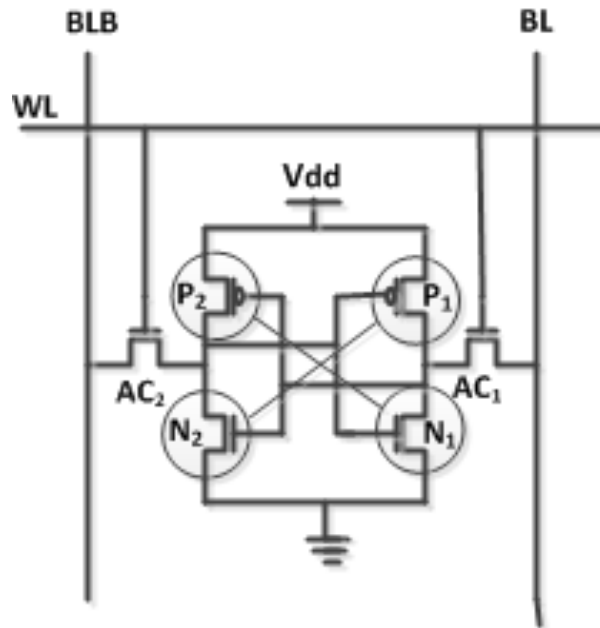


Figure 2.23: Periodic cell flipping

### 2.5.2.2 Supply Voltage Scaling

BTI degradation is very sensitive to supply voltage ( $V_{DD}$ ) because of its dependence to the vertical oxide field. Then, tuning  $V_{DD}$  dynamically when the memory cell is in the standby mode can mitigate some part of  $V_T$  shift during device operation. The result shows that  $V_{DD}$  scaling can significantly improve the cell stability under NBTI degradation mechanism [65].

Then, this technique has already been incorporated in memory architectures to minimize the leakage power consumption. Beside its benefit, it would also lead to slower memory access time as a drawback. The  $V_{DD}$  scaling approach efficiency depends on standby periods in the memory and in the cases that the memory cannot go to standby mode for long periods, the devices would experience permanent wear out which would decrease the memory lifetime and increase the probability of faults in the memory.

## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

### 2.5.2.3 Adaptive Body Biasing

The Adaptive Body Biasing (ABB) is a conventionally used technique to reduce the leakage power consumption in low voltage memories. Recently, it has been utilized to mitigate the impacts of process variation and BTI aging in memory cells (Figure 2.24) [17] [66]. As the main impact of BTI aging is on the threshold voltage of transistors, the adaptive body biasing approach can control the transistor threshold voltage by applying a voltage source to body bias ( $V_{BS}$ ) to the transistor. For instance, applying a Forward Body Bias (FBB) can reduce the transistor threshold voltage.

In principle, the ABB approach need separate voltages inside a die to bias each device independently, which leads to large area overhead. Therefore, it is more efficient to utilize on chip reliability sensors to monitor the degradation in memory cells and to apply the appropriate body to source voltage  $V_{BS}$  to the transistors globally, in order to mitigate some part of the threshold voltage shift and to remove potential failures from the memory array [66].

Although this global applied voltage body to source can reduce some part of threshold voltage shift in the memory cells, but having for instance 6 transistors in 6T SRAM cell in which their threshold voltage shift may vary according to cell signal probability and device variability, decreases the reliability enhancement efficiency by applying just one global body to source voltage in the memory.

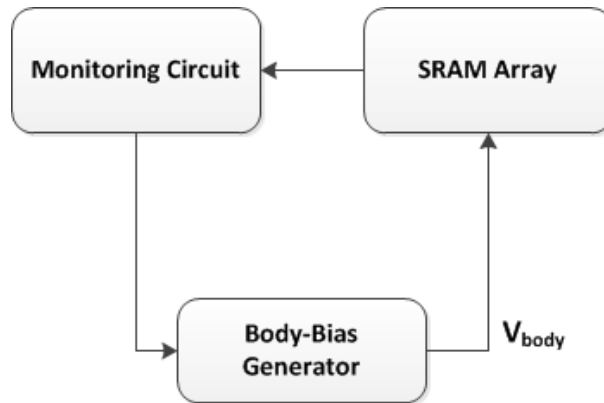


Figure 2.24: Adaptive body biasing in SRAM array

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

Note that, the adaptive body bias technique does not mitigate the permanent BTI aging occurring in the transistors since the memory cells does not experience any recovery time during their operation, and it is just their body to source voltage that varies time to time. Moreover, with technology scaling down the sub 22nm sizes not the body effect coefficient decreases and that result in smaller impact of body biasing technique for mitigating aging effects.

Finally, note that, the technique of ABB, currently, cannot be applied in the advanced FinFET technology transistors, as they do not contain any body contact, therefore, new mitigation techniques are needed for them, techniques such as the proactive reconfiguration, to be studied in the following.

### 2.5.2.4 Proactive Reconfiguration Strategy

Proactive reconfiguration is based on utilizing the available redundancy dynamically in such a way that the spare part replaces the operative parts in order to extend the overall system lifetime, while the reactive reconfiguration utilizes the redundancy only when a failure happens.

The proactive reconfiguration concept was introduced by IBM in 2008 [33] as a technique to improve reliability in SRAM memory arrays. Its benefit is to extend the system memory lifetime, thanks to the use of the spare parts in the normal operation of the memory system. In this strategy, redundancy is used in such a way to permit the operating elements go to a recovery period well in advance their fail. This would allow active elements to operate in activated and deactivated modes in a rotating basis, based on a recovery schedule, and therefore recovers some part of the BTI aging effects.

The recovery from aging results is mitigating some part of SRAM cell stability deterioration. Moreover, proactive reconfiguration has significant benefits over the reactive (conventional) reconfiguration, even if a limited number of redundant elements are used proactively. As it enlarges the lifetime (time until the final failure due to system aging) of working elements by suspending them from their active mode and mitigating some part of their



## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

---

wear-out, specifically for samples subjected to BTI stress, which presents recovery properties. Otherwise, in the reactive reconfiguration the redundant elements can only replace the failed elements and it does not provide any facility to ward off the element's aging. Therefore, while in the reactive reconfiguration the number of faults over a period of time increases along the lifetime for instance because of aging, proactive reconfiguration can extend the lifetime by suspending the units operation and putting them in recovery mode. Moreover, with proactive reconfiguration the elements degradation will be distributed and balanced between all of them.

One of the main considerations in proactive reconfiguration is the recovery management. The fact that how long and how frequent putting the units in recovery mode can impact the efficiency of proactive concept. Recovery can be applied when the system is in idle and by utilizing an obvious method such as a round robin over regular time intervals. However, using more intelligent recovery approach can enhance the efficiency of proactive technique. Moreover, the recovery mechanism can be accelerated by applying some reverse bias voltages to devices during the recovery phase [67].

Proactive reconfiguration in memories can be implemented in different granularity levels such as memory arrays, memory rows or columns. Figure 2.25 shows an example of proactive reconfiguration between the SRAM memory arrays. This is an example for simultaneous utilization of functional and spare units in the memory, where the memory is constructed with arrays.

It is an 8-way set associative cache consisting of 64 arrays, eight of which compose one associative way (each row in the figure). It also contains one additional spare array to be used proactively to allow any of the 64 arrays to operate in recovery mode at any given time. If this spare array was used reactively, then, it could just replace one failed array at a certain time and other arrays could not experience any recovery mode.

As an example, it is considered that the recovery mode of arrays is a round robin scheduling. Therefore, the arrays are replaced one by one by the spare array and go to recovery mode. First, the array  $a_{11}$  data is written to the  $a_0$  spare array and then the  $a_{11}$  array goes to recovery mode. After  $a_{11}$  finishes its recovery time, it transitions out of the recovery mode. The

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

round robin moves forward and it is the time for the next array  $a_{12}$  to enter recovery mode and to mitigate some of its degradation. Now the  $a_{12}$  data is copied into  $a_{11}$  and the  $a_{12}$  array is deactivated and enters into recovery. As the round robin continues all of the arrays will experience recovery once.

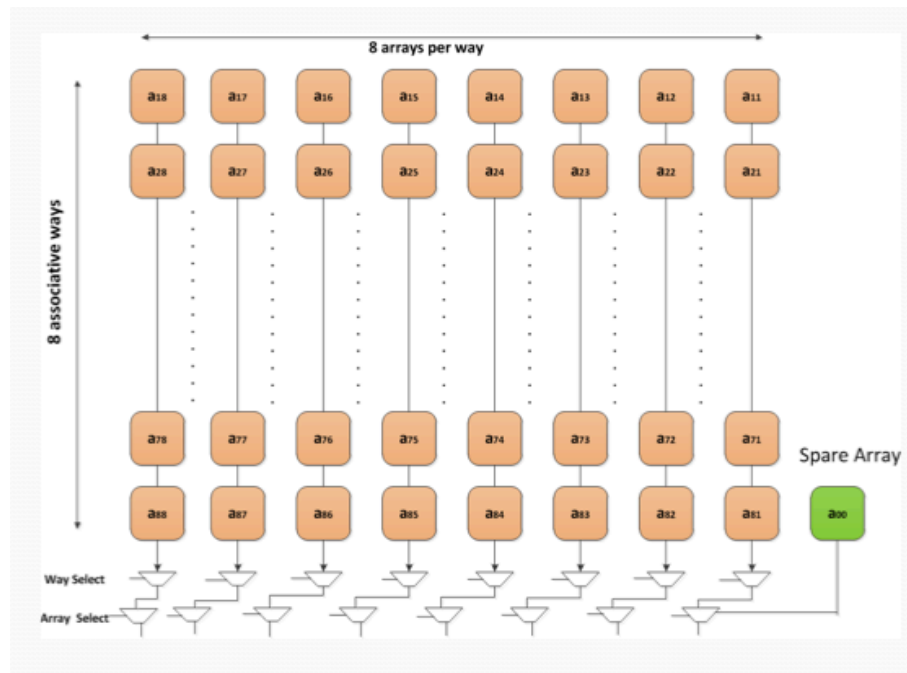


Figure 2.25: A block diagram of memory array for proactive reconfiguration [33]

### Comparison of Proactive Reconfiguration with other repair techniques

In [33] the proactive reconfiguration was compared with other conventional approaches, such as reactive sparing and ECC. The comparison was in terms lifetime extension and area overhead. Figure 2.26 presents the lifetime extension using ECC method, column spare and row and array spares.

It is observed that the ECC method and reactive use of spare column, row and arrays cannot extend the lifetime more than 3 times even with increase in the area overhead, while the proactive reconfiguration (using only one spare array proactively) can extend the memory lifetime up to 7 times on average.

## CHAPTER 2. STATE OF THE ART OF MODERN AND FUTURE TECHNOLOGIES FOR COMPUTER MEMORIES

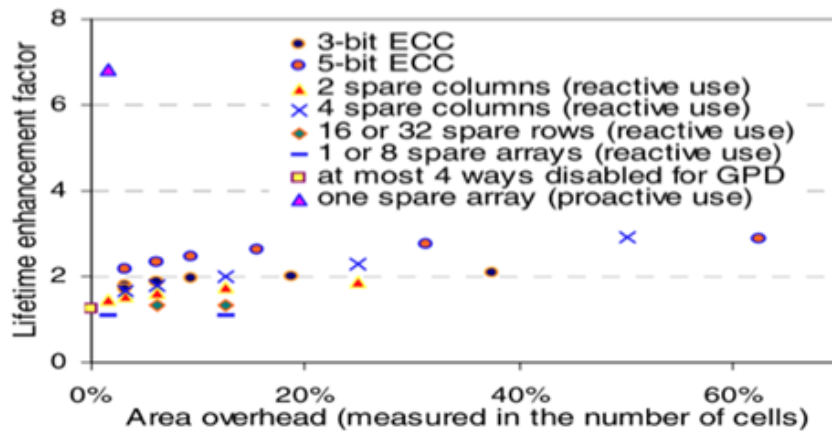


Figure 2.26: Lifetime extension versus area overhead of conventional techniques and the proactive reconfiguration [33]

Furthermore, performance analysis results show that proactive reconfiguration has smaller performance losses than the other repair techniques. Note that another advantages of proactive reconfiguration is its flexibility such that it can be utilized in conjunction with other approaches such as the reactive techniques.

In summary, proactive reconfiguration approach has significantly better lifetime-area and lifetime-performance trade-offs than the other conventional methods and it is a good candidate for reliable memory design. Note that, the repairing efficiency of proactive reconfiguration would be higher when it is used in chip failure mechanisms that could benefit from recovery properties (such as in BTI failure mechanism). For his we have considered proactive reconfiguration as the baseline technique to accomplish this thesis.



# 3

## Process Variability and Reliability Concerns in CMOS Technologies

### 3.1 Introduction

As the device technology nodes scale down 45nm and beyond, one of the major threats and source of unreliability and drop of performance in nano-scale circuit design is the higher relevance of the different fluctuation sources [28]. These sources of variation would affect a wide set of the transistor model parameters and the robust operation of the circuits.

Variability in integrated circuits can be categorized to two main areas: 1-Time-zero variations (TZV), 2-Time-dependent variations (TDV). These two induce parametric variations, being one of the transistor parameters that is significantly affected by the variability phenomenon, and considered with special attention in this thesis, is the threshold voltage ( $V_T$ ).

### 3.2 Time-zero Variability

Spatial variations, or variations that exist just after the manufacturing in the fresh samples, fall into two main category of fluctuations [17], the inter-die variation and the intra-die variations. One of the major sources that causes this type of variations origins from imperfect optical lithography during the fabrication. The present nano-scale design technology is still utilizing the 0.193um sub-wavelength, which is bigger than the device size (for instance as technology node scales to 45, 32, 28nm), and therefore inducing sources of variation (mainly line edge roughness) at time zero [17].

Figure 3.1a shows a statistical distribution of  $V_T$  variability in a die, while Figure 3.1b depicts an example of time zero variability for  $V_T$  in different gate length technologies.

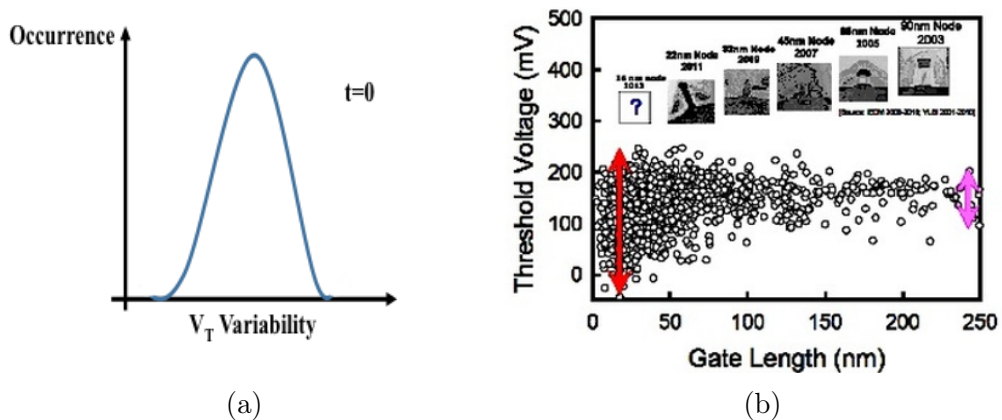


Figure 3.1: a) An example for  $V_T$  variability at time zero, b)  $V_T$  variability at different gate lengths [68]

#### 3.2.1 Inter-die Variations

Inter-die variation generally results from different runs of manufacturing between wafers, and could be because of process fluctuations in length, width, oxide thickness and etc.

In this type of variability the transistors parameters such as the  $V_T$ , all would be in the same direction (either high or low) inside a die. This type of

## CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

---

variation can be better predicted by knowing the location on the wafer or die and its impact is captured by using random variables through using corner models [69].

### 3.2.2 Intra-die Variations

The intra-die variation is the deviation in transistor parameters that exist inside the same die. There exist systematic and random sources for intra-die variability, where most of the systematic ones have been mitigated in previous research works (such as regular layout design). Therefore, the main sources of random intra-die variation include: 1-Random Dopant Fluctuation (RDF) 2-Line Edge Roughness (LER) presented in the following and considered in this thesis.

#### **Random Dopant Fluctuation (RDF):**

The fluctuation in number and location of dopant atoms in the transistor channel is defined as RDF [70]. In old CMOS technologies the number of dopants inside the channel was big enough and the variation in their number and location was not critical. However, in modern CMOS technologies, for instance below 45nm, the number of dopants has decreased dramatically (less than 100 atoms in 32nm [52]) and therefore the RDF has become the most significant source of variation in transistor parameters such as  $V_T$  [6].

As an example simulation results have shown that RDF is the reason for  $\sim 60\%$  of the total PMOS  $\sigma V_T$  in 45nm technology node [6]. Figure 3.2 shows an example of RDF effect inside transistor [70], where each small dot represents one dopant atom and it is seen how small they are in comparison with the channel dimensions. Note that, RDF has been one of the major reasons for changing the transistor shape to FinFETs in which the  $V_T$  variation due to RDF is negligible [18] [71].

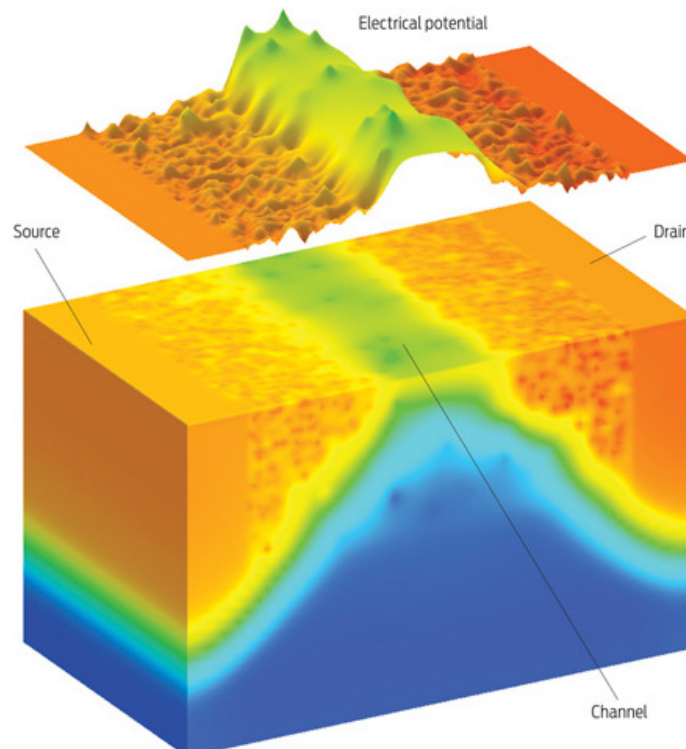


Figure 3.2: Random Dopant Fluctuation (RDF) in sub-micron CMOS technologies [70]

### **Line Edge Roughness (LER):**

The variation in poly width that constructs the transistor is denoted as LER [72]. This would cause mismatches between two transistors in the same die. LER has become important from technology nodes of 100nm, but it is even more significant issue of variability below 50nm [72].

LER is caused by a number of statistically fluctuating effects at these small dimensions such as shot noise (photon flux variations), statistical distributions of chemical species in the resist such as photoacid generators, the random walk nature of acid diffusion during chemical amplification, and the nonzero size of resist polymers being dissolved during development. It is unclear which process or processes dominate in their contribution to LER [72]. Figure 3.3 depicts an example of LER effect inside one integrated circuit. Figure 3.4 demonstrates that in FinFET technologies the LER can be seen as Gate Edge Roughness (GER) and Fin Edge Roughness (FER). It is stated



### CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

that the variations due to LER is the main source of  $V_T$  variation for 10nm SOI FinFET technology [18].

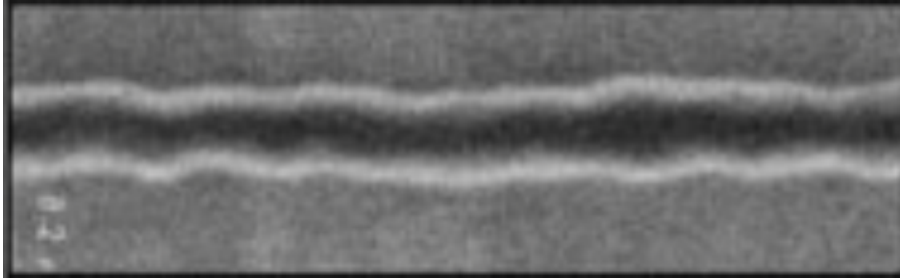


Figure 3.3: LER in sub-micro CMOS technologies, [72]

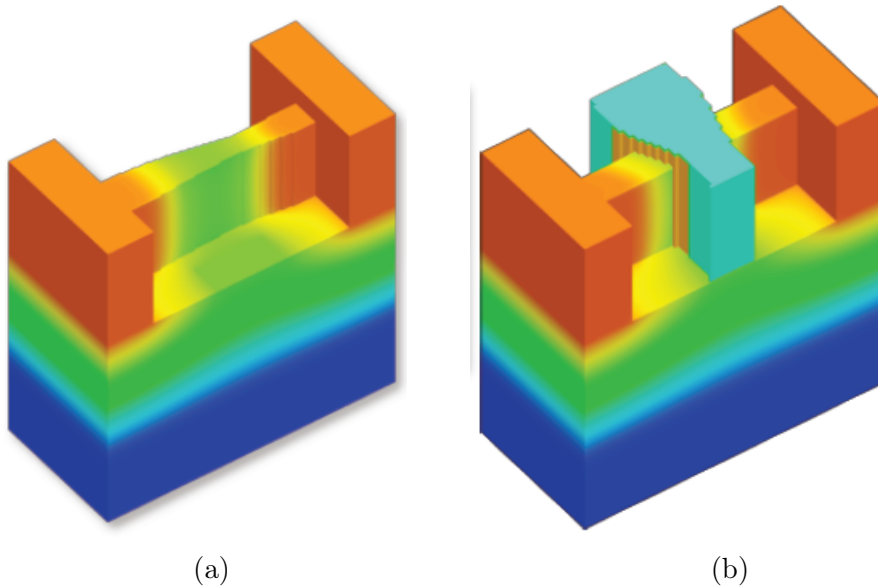


Figure 3.4: a) Fin Edge Roughness(FER), b) Gate Edge Roughness(GER), [18]

These variations (RDF and LER) pose a significant attention and design considerations when the devices scale down to nano-scale sizes of sub-22nm [35] as the levels of variability in the threshold voltage may arise to levels of the standard deviation of the distribution around a 35% of the average value [73], causing an important yield drop.

Figure 3.5 shows the current-voltage relation in 13nm PMOS and NMOS transistors in 1000 samples, and the dotted lines are the average. It is ob-

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

served there is a big drift between the current and voltage relation, which can result failures in integrated circuits.

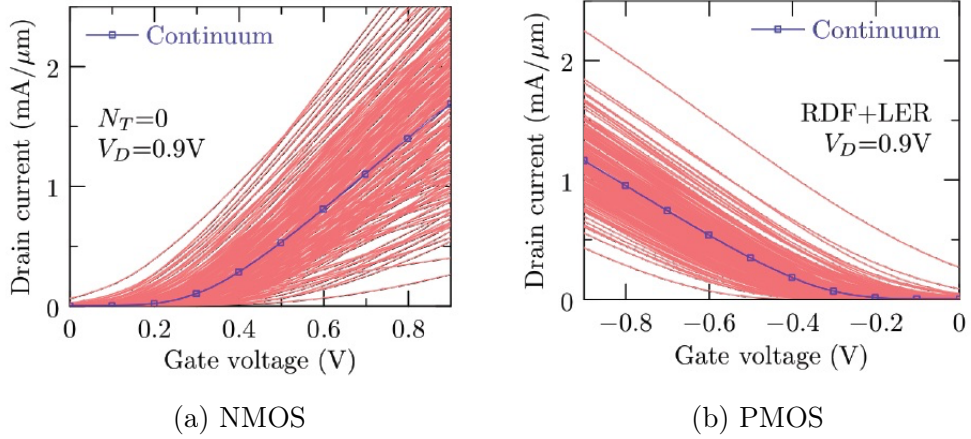


Figure 3.5: Process variability in nano-scale CMOS devices [35]

To summarize, time-zero variability components are classified into systematic, i.e. affecting a given region of the integrated circuits in the same way (mainly due to photolithographic distortions) and pure random, i.e. affecting the device independently. From all the above, then we can assume that in integrated circuit, all transistor  $V_T$  values will exhibit a statistical distribution in such a way that they are given by the two components in Eq. 3.1:

$$V_T(0) = V_{T, \text{systematic}}(0) + V_{T, \text{random}}(0) \quad (3.1)$$

where (0) indicates the  $V_T$  values at the origin of time just after the manufacturing process.

### 3.3 Time Dependent Variability

This type of variation is the fluctuations in transistors parameters due to aging related mechanism during their working lifetime. The main aging mechanisms include the Bias Temperature Instability (BTI), Hot Carrier Injection (HCI) and Time-Dependent Dielectric Breakdown (TDDB).

In this thesis BTI is considered as the most significant aging mechanism in Bulk-CMOS technologies, and will be modeled for simulations. Note that, there exist another type of variation called environmental variation, which include the voltage and temperature deviations, however this thesis will only consider their impact in BTI aging and will not analyze these types of variations independently. Figure 3.6 shows a statistical distribution of  $V_T$  variability in a die getting shifted along the lifetime, where  $t_1$  and  $t_2$  mean the time at different periods of operation.

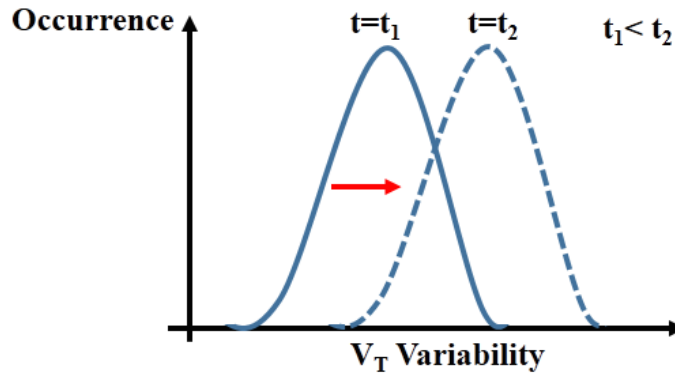


Figure 3.6: An example for  $V_T$  variability shifting in time

#### 3.3.1 Bias Temperature Instability (BTI)

The BTI degradation phenomenon known as Negative Bias Temperature Instability (NBTI) in PMOS transistors and Positive Bias Temperature Instability (PBTI) in high-k NMOS transistors has become one of the major

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

reliability concerns in nano-scale Very Large Scale Integration (VLSI) design [74][75].

NBTI degradation is consequence of continuous trap generation in  $Si - SiO_2$  interface of PMOS transistors and Figure 3.7 exhibits one example for this mechanism. Generally, and in order to transform hanging Si atoms to Si-H bonds, hydrogen is applied to the Si surface after the oxidation process. During the device operation, and when the PMOS device is under bias (i.e,  $V_{gs} = -V_{DD}$ ), Si-H bonds can break, due to positive holes in the channel. In this phase, the H atoms diffuse to the gate and the accumulated broken bonds act as interfacial traps capturing the electrons flowing from source to drain eventually increasing the device threshold voltage ( $V_T$ ). The corresponding phase is called the stress phase.

If the device is released from the stress (i.e,  $V_{gs} = 0$ ) then some of the H atoms will diffuse back and some of the Si-H will form again. This would reduce the number of interface traps; therefore some part of degradation would be recovered. The corresponding phase to this mechanism is called the recovery phase.

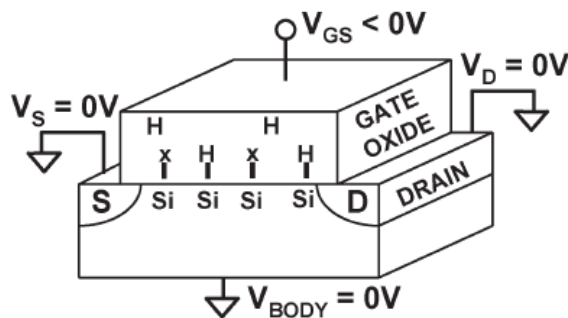


Figure 3.7: NBTI aging mechanism in PMOS transistors [75]

In older CMOS technology ( $>45nm$ ) the PBTI effect that damages the NMOS behavior was not a serious reliability concern, since the number of holes in the channel of NMOS transistor are negligible and therefore very few interface traps were generated. However, PBTI is observed to be an important issue because of replacing  $SiO_2$  dielectrics with high-k dielectrics [4]. Then, this is caused by filling of pre-existing defects by electrons tunneling

## CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

from the channel into the gate dielectric bulk [4].

So then, both NBTI and PBTI result in shift of  $V_T$  along the device lifetime and Figure 3.8 shows an example for  $V_T$  increase in TiN and Re gated devices with  $SiO_2/HfO_2$  as dielectric stack (at 45nm technology node) in respect to the device stress time. Experiments show that the three main factors that affect the BTI and cause  $V_T$  drift in devices are voltage, temperature and duty cycle (defined as the percentage of period that the state stays high). However, the impact of frequency at BTI is independent, or weakly dependent [76][77]. Therefore, higher voltage, higher temperature and longer stress time, result in higher degradation [78].

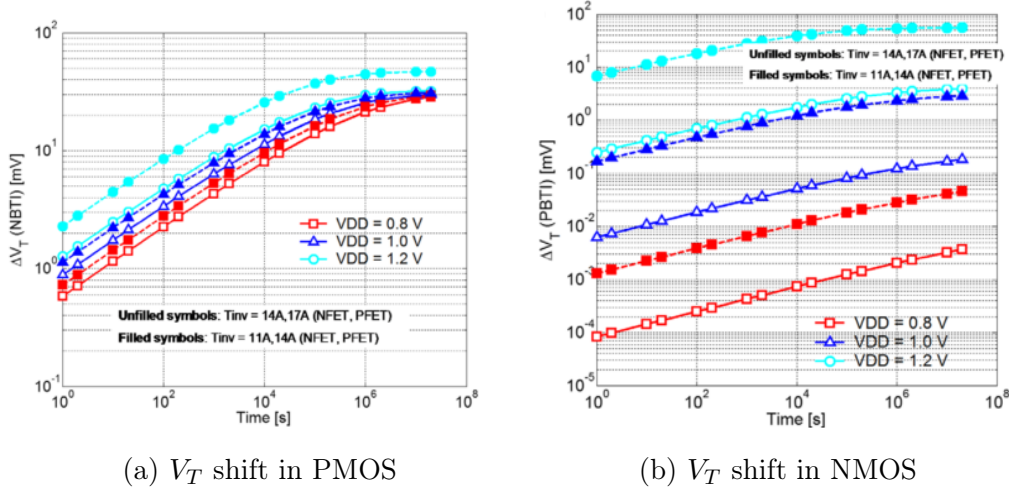


Figure 3.8:  $V_T$  shift due to NBTI and PBTI in PMOS and NMOS transistors [79]

Several models have been described to predict the impact of BTI aging on  $V_T$ -shift [76][80]. For instance, one of the models to analyze the BTI degradation phenomenon is the empirical Reaction-Diffusion (RD) framework. This model [75][81] interprets the phenomenon as equation Eq. 3.2, which makes it possible to evaluate the effects of BTI at circuit level.

$$\Delta V_T \propto \frac{qN_{IT}(t)}{C_{OX}} \propto f_{AC}(S_p) \times K_{DC} \times t^n \quad (3.2)$$

where  $N_{IT}$  is the generated interface traps,  $C_{OX}$  is the oxide capacitance,

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

$K_{DC}$  is technology dependent factor also related with temperature,  $f_{AC}$  represents the AC dependency of the process which is approximately linear [81],  $t$  which is the stress time and finally  $n$  the constant depending on H diffusion. Regarding this, Figure 3.9 exhibits an AC RD model for a BTI degradation mechanism.

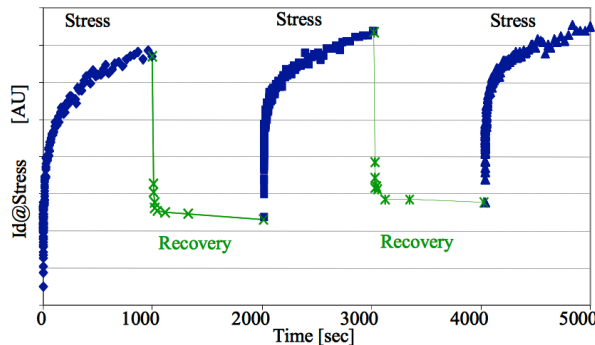


Figure 3.9: PMOS degradation and recovery behaviour [82]

Note that, BTI impact gets even worse in scaled technology due to the higher operation temperature and the usage of ultrathin gate oxide (i.e., higher oxide Field) [83]. However, one of the important characteristics of this aging mechanism is its recoverability presented in next section.

### BTI Recovery

It has been demonstrated the BTI is a sum of two damage components: a recovering and permanent [83]. The recovering component is the amount of degradation that is annealed just after removing the stress voltage, meanwhile the permanent component is the degradation that remains unchanged also when the device is not stressed.

Figure 3.10 shows these two components and their overall degradation impact in a NMOS transistor. BTI recovery process occurs just immediately after the stress voltage is removed, and during this phase some part of the  $V_T$  shift is mitigated. Figure 3.11 shows an example of  $V_T$  decrease during the recovery phase. Furthermore, applying a voltage inverse to stress voltage can speed up the recovery process in P/NMOS devices [83]. In Figure 3.11 recovery behavior is accelerated when a moderate positive bias is applied to a PMOS transistor.

### CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

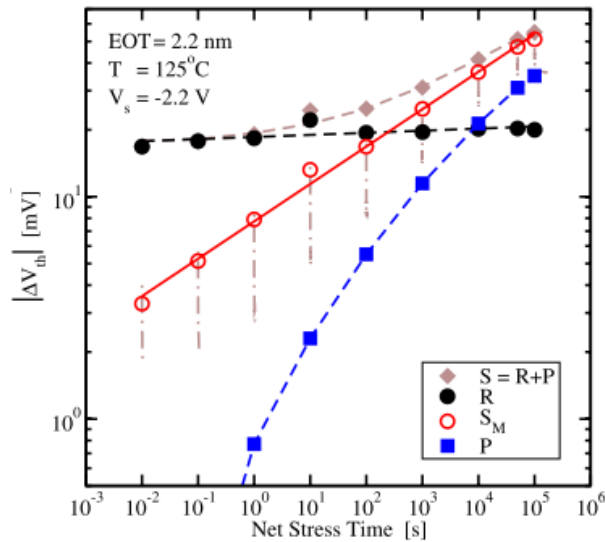


Figure 3.10: The recoverable (R) and permanent (P) damage components [83]

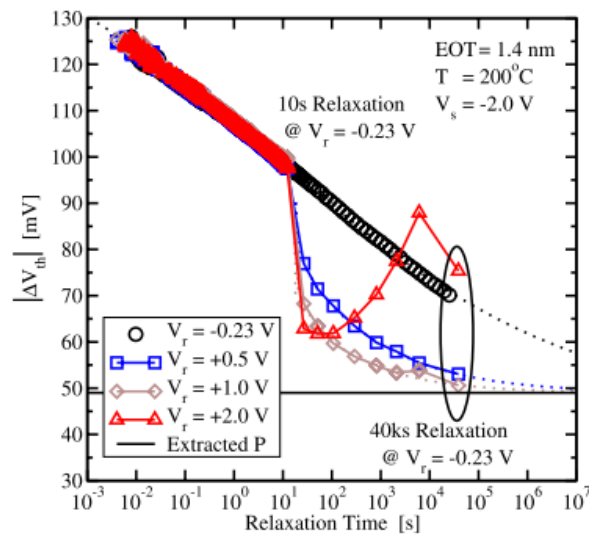


Figure 3.11: Applying a positive bias to the PMOS transistor accelerates the degradation recovery [83]

It is observed that giving the device enough relaxation period (i.e.  $10^4$  seconds in this example), can result in a decent percentage of degradation recovery and would mitigate some part of the  $V_T$  drift.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

### Bias Temperature Instability in FinFETs

Similar to planar CMOS, FinFET devices are also affected by BTI aging and their  $V_T$  shifts during the device stress time. BTI induced  $V_T$  shift is fitted by a power law in Eq. 3.3 based on wafer-level extended Measure-Stress-Measure (eMSM) measurements on commercial-grade 28nm HKMG and research-grade FinFET nodes at 125°C [84].

$$\Delta V_T(t) \cong At^\alpha E_{OX}^\gamma \quad (3.3)$$

where  $t$  is the time,  $A$  is the fitting coefficient,  $E_{OX}$  is the electric field across gate oxide and  $\alpha, \gamma$  are the acceleration exponents for the electric field across gate oxide. Figure 3.12 shows the absolute shift of the  $V_T$  for a 14nm pFinFET at 125°C under the time and voltage acceleration.

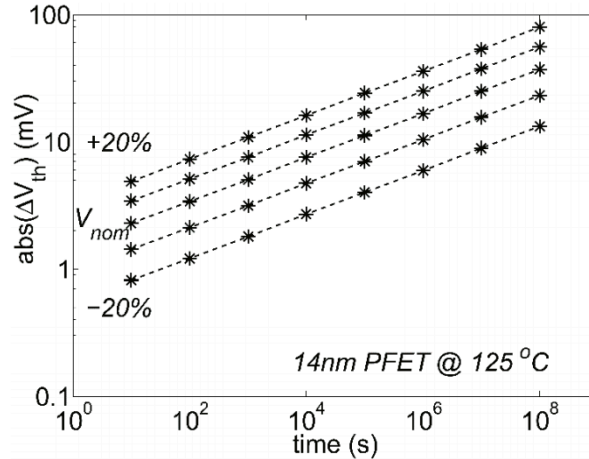


Figure 3.12:  $V_T$  shift due to BTI under time and voltage acceleration in 14nm pFinFET [84]

It has been shown that BTI recovery in tri-gate devices matches data and model predictions from planar devices [85], also just as in planar devices increasing temperature can enhance the recovery in tri-gate devices [86]. Regarding these, Figure 3.13 shows the recovery characteristics between planar (20nm) and FinFET devices (14nm).

Note that unlike planar devices BTI recovery in FinFETs depends much more on the stress time than the voltage amplitude and higher stress time



### CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

will have higher impact on aging than the higher supply voltage [23], as it is shown in Figure 3.14.

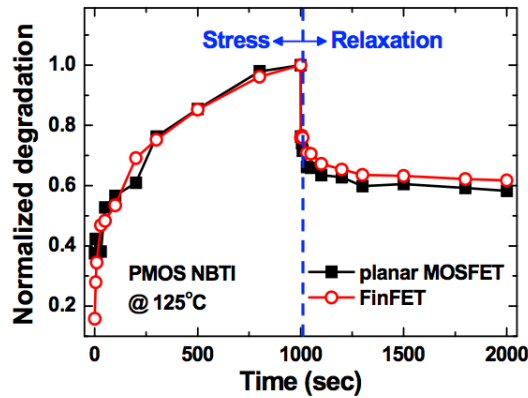


Figure 3.13: BTI recovery characteristics for PMOS NBTI in both 20nm planar and 14nm FinFET show similar recovery properties [85]

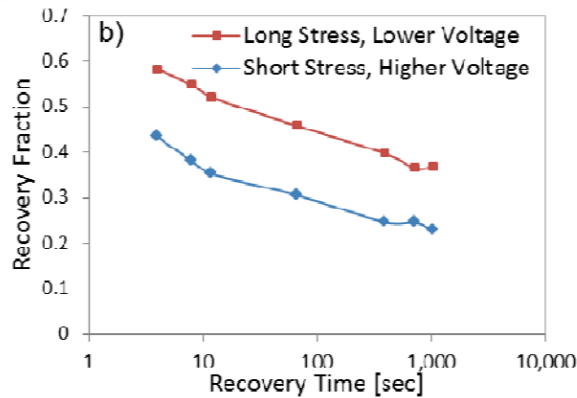


Figure 3.14: BTI recovery in FinFETs is more affected by stress time, devices with shorter stress time recovered faster [86]

All these signify the importance of BTI aging in advanced FinFET devices and motivates a strategy that can benefit from recovery property in these devices.

### 3.3.2 Other Reliability Concerns in CMOS Technologies (HCI, TDDB)

In addition to BTI aging (studied in section 3.3), there exist some other failure mechanisms in modern CMOS devices. These mechanisms can be categorized into two types, one is the group related with aging and includes Hot Carrier Injection (HCI), Time-Dependent Dielectric Breakdown (TDDB), while the other is not time dependent and include Electro migration, Random Telegraph Noise (RTN). In this section only the aging related mechanisms are presented.

#### HCI

Hot Carrier Injection (HCI) is an aging mechanism, which has also become a major reliability concern in modern CMOS technologies [21]. It was expected that HCI would become less important with the technology scaling and more specifically the reduction of voltage supply, however due to increase of internal electric fields inside the nano-scale device its effect has recently gained significant attention.

This type of aging happens when an interface trap is created inside the pinch off region (drain side), because an electron or hole obtains enough energy to be transported into the gate dielectric. The high electric field near the drain side of the device heats the carrier and causes the effect, which results into impact ionization and wear-out. The result is deviation in device parameters such as its  $V_T$  that would consequently reduce the device performance along the lifetime similarly to the BTI aging.

Generally, the HCI is more severe in NMOS devices in comparison with PMOS, because the electrons of the channel have higher mobility and consequently can absorb more kinetic energy for tunneling. HCI has direct relation with the frequency and therefore higher frequency would lead to faster HCI aging inside the device.

#### TDDB

Time-Dependent Dielectric Breakdown (TDDB) is a degradation mechanism that occurs inside the device because of applying high electric field to its gate

## CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

---

dielectric [22]. Generally a conductive path is formed between the gate and the substrate, which may shorten the anode and cathode. This conductive path will let that some current to flow between gate and substrate and would cause thermal heating inside the device.

TDDB has a direct relation with device scaling, as the device scales down to smaller technology nodes, the gate dielectric thickness gets thinner and therefore the impact of electric fields would be stronger. Note that TDDB can cause two types of failures, a soft breakdown if the device continues to function, and a hard break down if the dielectric breaks completely, making the device out of functionality.

Unfortunately both the HCI and TDDB do not benefit from recovery properties like the BTI aging, therefore to mitigate their effect some recovery independent design techniques would be needed.

### 3.4 Time-zero Variability and BTI Aging Modeling in This Thesis

In this section, the main objective is to define an approach of how to take into account the impact of time zero variability and aging in reliability analysis of memory cells. In this context, a linear-based approximation technique is introduced to model the BTI degradation for memory lifetime analysis. Regarding this, one of the conventional approaches in studying the impact of time zero variation for circuit analysis is to consider them as statistical distributions. Therefore, for example, time-zero variations are defined as normal distribution (where  $\mu$  =mean and  $\sigma$ =standard deviation of process parameters). In this thesis and in order to analyze the time-zero variability,  $\pm \frac{3\sigma}{\mu}$  of parameter spread around its mean value is considered. For instance, a defined value is considered as the mean value for  $V_T$  ( $\mu V_T$ ), and then  $(\mu V_T - 3\sigma) \leq V_T \leq (\mu V_T + 3\sigma)$  is the distribution of values for the simulation analysis in this thesis.

Next, and in order to model the Bias Temperature Instability (BTI) aging mechanism, its main behavior is considered from a circuit perspective of view,

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

where it mainly leads to a  $V_T$ -shift in transistors along their working lifetime. Therefore, this thesis mainly seeks an approach to model this aging behavior for circuit and system analysis by using numerical simulations. Since this work implies an analysis during a long period of time, the aging model is simplified to linear equations.

First, let's analyze the BTI aging in the stress phase. It is observed in the stress phase, that the device suffers some  $V_T$  shift due to specific physical mechanisms affecting materials. Previous works have demonstrated that the  $V_T$ -shift during the device lifetime is sub-linear with a fast aging slope at the beginning, and a posterior slower slope [79].

Figure 3.15 illustrates how the BTI aging stress phase can be modeled with piece-wise linear slopes, with an initial sharp increase at the beginning of stress phase and some decreasing slopes in the following of the lifetime. The  $K(t)$  factor is the time-varying aging slope ( $K(t)$  differs in each time period, because the aging impact is time-dependent) and the respective values for further analysis will be taken from previous experimental measurements [79] [80].

Once the device is released from stress, the recovery phase starts. This may mitigate some part of the  $V_T$  shift (Figure 3.16), and as a consequence can extend the device lifetime. This wear-out recovery is modeled in respect to different possible technologies.

The BTI wear-out magnitude can depend among others on the device environment, e.g. temperature and voltage [76]. To model all these effects, a parameter is introduced that we called it Recovery factor ( $Rf$ , expressed in percentage).  $Rf$  is defined as: The proportion of  $V_T$ -recovery (reduction of  $V_T$ ) regarding the previous stress phase and its corresponding degradation.

CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

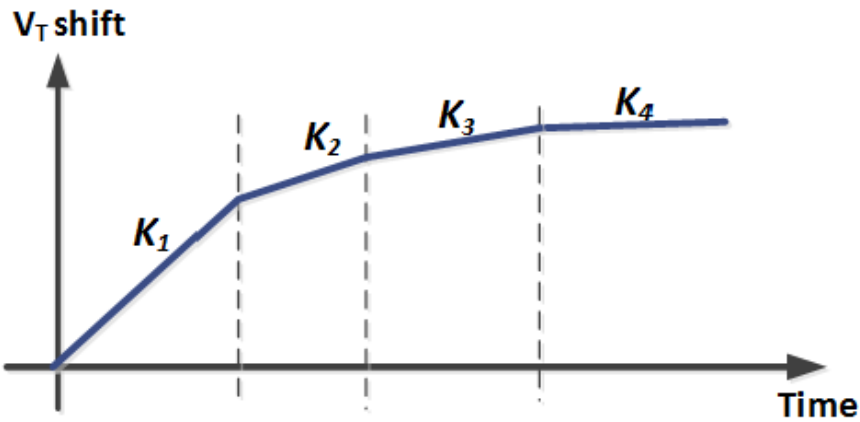


Figure 3.15:  $V_T$ -shift evolution when samples are subjected to BTI stress. The stress time divisions are well defined

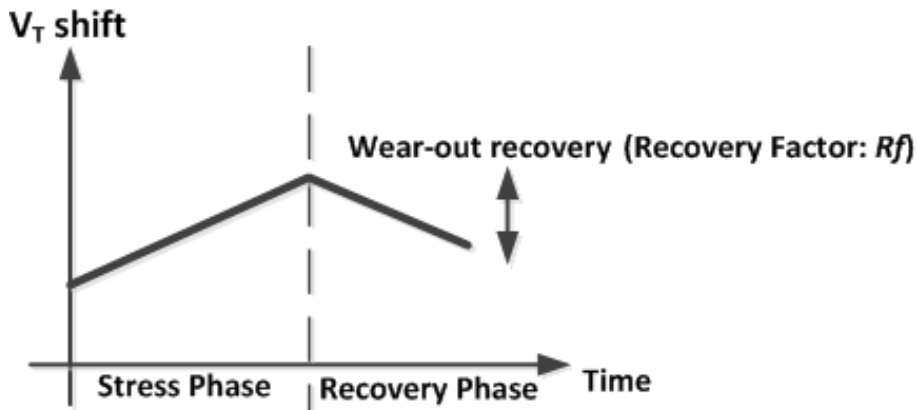


Figure 3.16:  $V_T$ -shift performance of a device subjected to BTI stress. The behaviour is divided into two phases: stress and recovery

For instance,  $Rf = 50\%$  means that half of the  $V_T$ -shift from the previous stress phase would be recovered after the relaxation phase (see Figure 3.17). Experimental measurements [83] have demonstrated that the device have a sharp recovery just after being released from being under stress, and later the recovery gets slow. A decent amount of wear-out recovery is obtained after a recovery time (TR) of  $10^4$  seconds. Therefore, in this study, it is assumed that the minimum time to optimize the BTI wear-out recovery is about  $10^4$  seconds (the recovery periods in this work are bigger than or equal to  $10^4$

seconds).

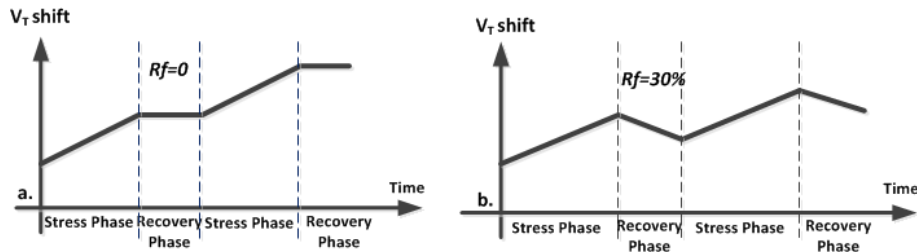


Figure 3.17: BTI wear-out recovery behavior with different recovery factors: (a)  $Rf=0$  and (b)  $Rf=30\%$ .

### 3.5 BTI and Process Variability in SRAM Memories

SRAM cells are built with very small sized devices making them prone to induced variability and BTI aging. Moreover, they may store the same data for long period of time (being read multiple times but not flipping the data), which might induce asymmetric device degradation. To analyze the impact of process variability and BTI in 6T SRAM cells some of the robustness metrics in SRAM cell are considered, e.g. Static Noise Margin (SNM) and writability in the SRAM cell [87].

SNM was defined in Chapter 2 as the ability to perform a correct read operation (Read  $SNM$ ), and writability of a cell is measured by how much the voltage at BL needs to be lowered to flip the cell [79].

Considering this, Figure 3.18 presents the impact of device variability in the read SNM of 1kB SRAM cells [87]. Here, the one-side SNM is shown, where  $SNM(L)$  is the square of the left eye of the butterfly curve and  $SNM(R)$  is the square of the right eye.

Note that, as the device variability increase with further scaling the SNM squares get smaller and the SRAM cell might become unstable.

### CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

---

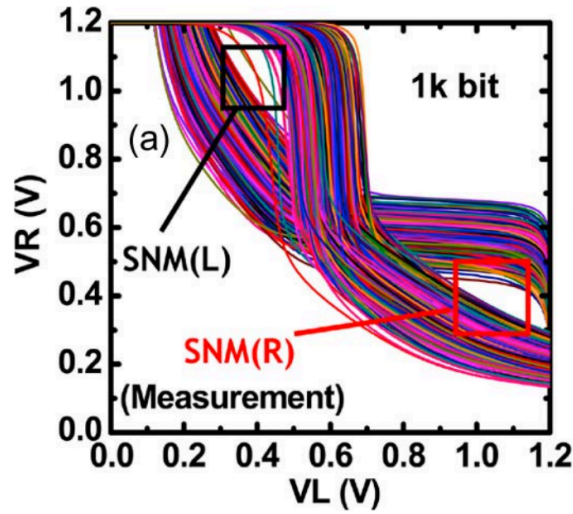


Figure 3.18: Measured butterfly curves of 1-kB SRAM cells [87]

Aging will also impact the SNM and for instance it is shown that SNM reduces linearly with  $V_T$  shift in P/NMOS due to both NBTI and PBTI, and their combined effect is additive. For instance, Figure 3.19 shows this SNM reduction in 45nm technology node 6T SRAM cells. It has also demonstrated that the simultaneous effect of NBTI and PBTI aging results in degradation of writability (marginally) in the SRAM memory cells [79].

Furthermore, [23] analyzes the impact of BTI aging in FinFET based SRAM cells. The result states that under nominal  $V_{DD}$  and for analysis of  $10^8$  seconds in operation the BTI causes Read  $SNM$  degradation of around 17%. It also states that the FinFET SRAM cells are more vulnerable to the BTI degradation (2X) in comparison with planar CMOS cells (22nm technology). Figure 3.20 compares the read  $SNM$  in 14nm FinFET and 22nm MOSFET based 6T SRAM cells under the nominal supply voltage.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

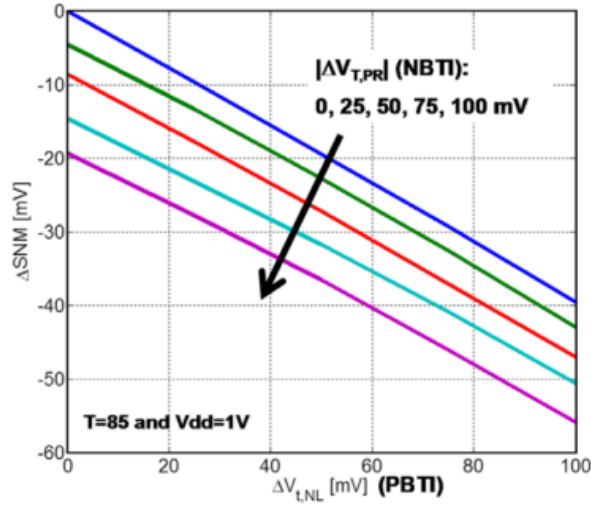


Figure 3.19: Combined effect of NBTI and PBTI on 6T SRAM  $SNM$  in 45nm [79]

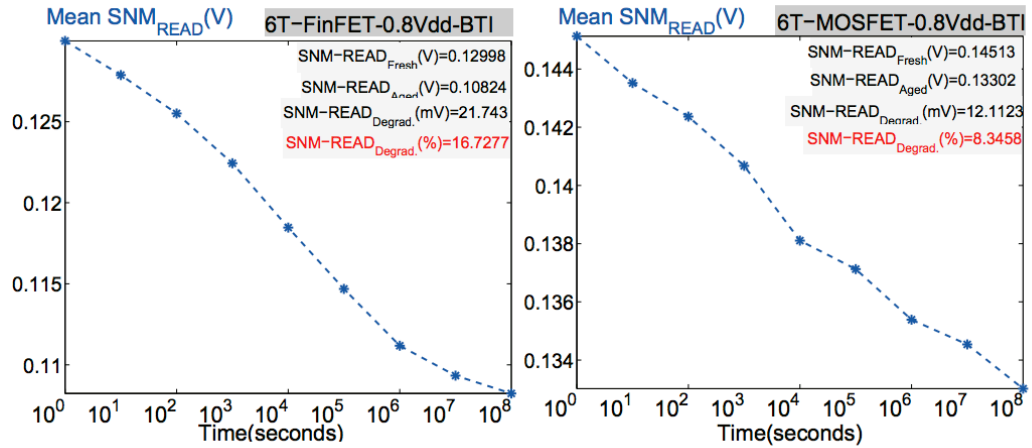


Figure 3.20: BTI induced aging in Read  $SNM$  for (a) FinFET and (b) MOSFET based 6T SRAM cells [23]

### 3.6 Summary and Conclusions

This Chapter of the thesis has analyzed the two types of variations (TZV, TDV) of device parameters, in modern CMOS technologies. Regarding this, first time-zero variability was considered and the main sources that cause this



### CHAPTER 3. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN CMOS TECHNOLOGIES

---

type of variability were described. Next, time-dependent variability (mainly BTI aging) was analyzed in nano-scale CMOS technologies.

As the main contribution of this Chapter, a modeling approach is considered in order to analyze the impact of time-zero variability and BTI aging in circuit and systems. The BTI aging is modeled with linear equations that emulate the  $V_T$  shift along the device lifetime and would be utilized to predict the final lifetime; this model will be used in the following chapters of the thesis to enhance the device lifetime through reconfiguring approaches.

Part of this Chapter, section 3.4 has been published in the Conference of VLSI Test Symposium (VTS 2012) [88].



# 4

## Proactive Reconfiguration Concept in SRAM Arrays

### 4.1 Introduction

This thesis focuses on the proactive reconfiguration principle to improve reliability in SRAM arrays. The concept of proactivity in SRAM arrays was first introduced in [33] as an approach to mitigate NBTI aging effects, later it was extended in [67] to both BTI aging in N and PMOS transistors of the SRAM cell. Its key benefit is to enlarge the system memory lifetime, thanks to the dynamic use of the spare parts in the normal operation of the memory system [33].

In the proactive context, the redundant elements (spare units conventionally utilized for yield enhancement) are used to allow the operating elements to enter in a well-established recovery period mitigating aging effect. This technique allows memory elements to operate in two modes: activated and deactivated, in a rotating basis, based on a given recovery schedule, and therefore able to recover some part of the aging effects. Moreover, proactive

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

reconfiguration has significant benefits over the reactive (conventional) reconfiguration technique even if only a limited number of redundant elements are used.

The application of the proactive mechanism enlarges the lifetime (time until the complete failure due to system aging) of the operative elements by suspending them from their active mode and mitigating some part of their wear-out, specifically for samples subjected to BTI stress, which presents recovery properties. Otherwise, in the reactive reconfiguration the redundant elements can only replace the failed elements and they do not provide any facility to ward off the element's aging. Additionally, with proactive reconfiguration basis the elements degradation can be homogeneously distributed and balanced between all of them.

Note that, the concept of homogeneous distribution of wear-out can be extended to a memory affected by process variability to make its elements age at different speeds by applying adaptive and intelligent recovery schedules. Proactive reconfiguration techniques can be implemented in different ways, depending on the redundancy granularity used. The hardware granularity level used in the following sections corresponds to the column elements in a memory system. The reason is related with their larger effectiveness to repair faults (i.e. bit line, sense amplifier, column multiplexer, column line decoding), and the greater fault coverage observed than for rows [61].

### 4.2 Analysis Framework

To figure out the efficiency of proactive reconfiguration in extending the lifetime of SRAM arrays a figure of merit is needed. Therefore, we have evaluated the two proactive approaches (IBM and the proposed reconfiguring approach in this thesis named as the adaptive proactive) with Matlab simulations [89] and compared the calculated lifetime from these two strategies with the non-proactive case. In this lifetime analysis the SRAM array is categorized by a number of columns, where each memory column is characterized by the cell with the largest  $V_T$  value (weakest component).

So then, the first proactive approach that we analyze is pointed out by

## CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT IN SRAM ARRAYS

---

IBM in [33], what corresponds to a basic homogenous proactive technique, and later we present the basis and implementation of an adaptive proactive technique originally proposed in this thesis.

To depict graphically the time-varying aging evolution and the lifetime behavior for these approaches, non-proactive, IBM proactive and adaptive proactive, we assume as a matter of example a system composed by four working columns and one spare column. In the presented example, a set of arbitrary fresh devices is stated for the worst  $V_T$  (the cell with lowest SNM) in each column (5 columns including the spare as number 5):  $V_{T1} = 320mV$ ,  $V_{T2} = 300mV$ ,  $V_{T3} = 290mV$ ,  $V_{T4} = 310mV$  and  $V_{T5} = 330mV$ , where  $V_{Ti}$  represents the  $V_T$  of the weakest cell for each column  $i$ . The maximum tolerated reliable  $V_T$  aging value ( $H$ ) before cell failure is assumed to be 400mV.

In this section, we compare three configurations, i.e. non-proactive, IBM proactive and this thesis proposed adaptive proactive technique, by considering different number of working and spare columns.

### 4.3 Aging Evolution in a Non-Proactive Scenario

We start the analysis by considering a SRAM array with a non-proactive reconfiguration scenario. In this scheme, the memory columns will experience BTI aging during all their lifetime, where we assume they will age by the same speed, and the lifetime will be when the  $V_T$  values exceed the assumed maximum  $V_T$  value ( $H$ ). Note that we consider the SRAM cells lifetime up to the moment ( $H$ ), when one column reaches by first time the maximum acceptable  $V_T$  value.

For instance, Figure 4.1 presents an example of SRAM structure having four operational columns and one spare column (SP1), where each column has 64 SRAM cells and is characterized by the highest  $V_T$  of the weakest cell. The architecture shows the five columns connected to 2to1 multiplexers where each one is controlled by the Finite State Machine (FSM), so that the

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

corresponding operative columns will be utilized in memory application.

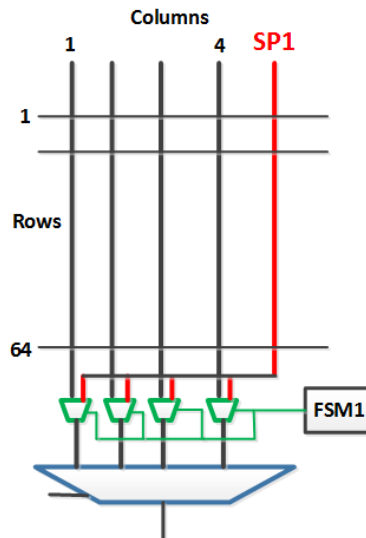


Figure 4.1: An example of a SRAM configuration with five columns (four operational + one spare SP1)

Afterwards, Figure 4.2 shows the corresponding system lifetime (84 months) for this non-proactive reconfiguration approach. The result shows that the lifetime depends on the column element with maximum threshold voltage value of the SRAM (here  $V_{T5}$ ), since a parallel behavior evolution is observed between them (same speed aging) and the worst column is the one that first arrives at the system failure criterion value. Moreover, this parallel evolution of the  $V_T$ -aging in this configuration needs to be mitigated in respect to the worst  $V_T$  cell in order to improve the SRAM lifetime.

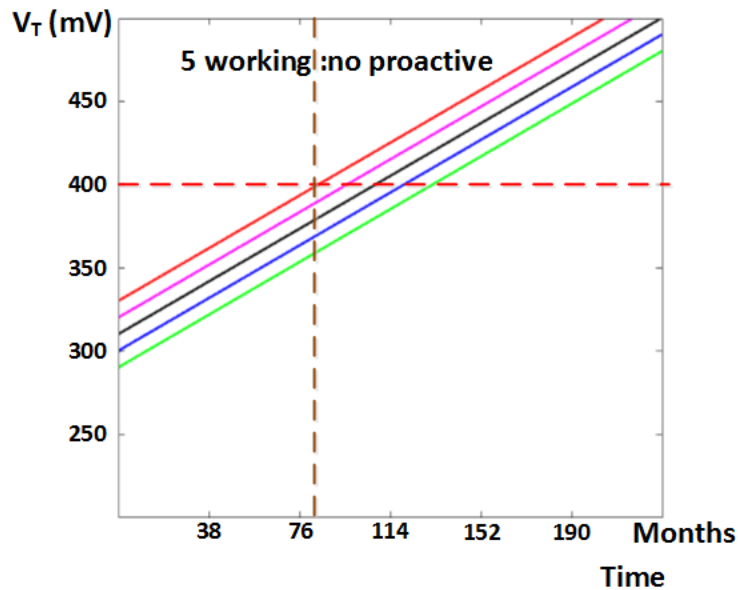


Figure 4.2:  $V_T$  aging slopes of columns in a non-proactive approach in a SRAM configuration of five columns

#### 4.4 Aging Evolution in the IBM'S Proactive Reconfiguration Technique (Homogeneous Approach)

On the other hand, the IBM's proposal is based on a time-homogeneous round robin strategy between memory columns (including spare ones), where all the columns go to recovery mode homogeneously, one by one for example in a rotating schedule [33] without taking into account the columns differential conditions (inherent process variability and during the time aging). This causes the elements to operate in activated or deactivated modes, and therefore allowing the recovering of some part of the BTI aging effects.

Figure 4.3 depicts the benefits of using this technique with a recovery factor  $R_f=30\%$ , in comparison with a non-proactive approach (where the system units are always under stress) in terms of system lifetime. We observe that for the proactive reconfiguration the aging slope of the transistors are

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

different causing an improvement of the system lifetime about 1.8X (150 months in front of 84). This example clearly shows the benefits of using the proactive reconfiguration approach.

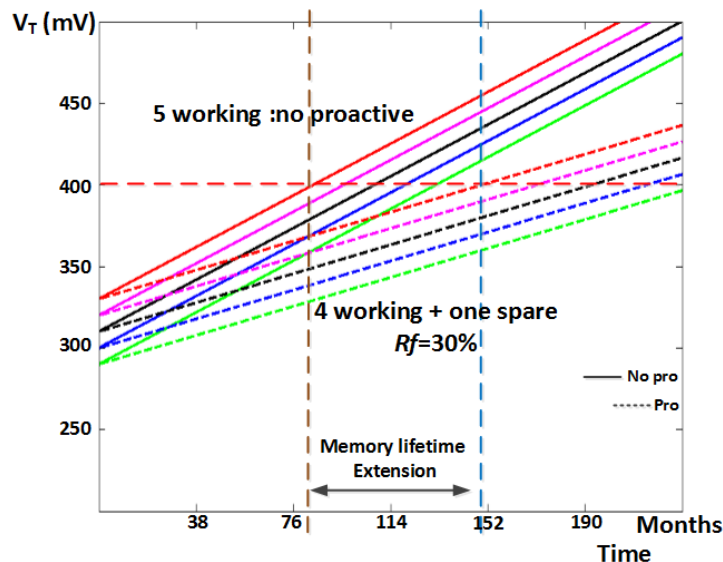


Figure 4.3:  $V_T$  aging slopes of columns in a proactive (dashed lines) and a non-proactive approach (continuous lines). Almost 2X lifetime extension is obtained.

In both non-proactive and IBM approaches the system fails when the weakest column (highest  $V_T$ ) arrives to the maximum acceptable  $V_T$  value. This occurs when there could exist other un-wasted memory columns that their memory cells are not fully aged. The columns age ( $V_T$ -shift) equally, (aging is slower in the IBM approach) which is presented in the figure by parallel slopes. Observe that, for the lifetime comparison it is assumed that both strategies are using all resources, but in different manners. Note that, the IBM proactive reconfiguration approach uses equals recovery periods, and can result in a relevant memory lifetime extension. Nevertheless, equal recovery periods of units do not mitigate the relevant time zero process variations of the deep transistors in SRAM cells. In this sense, the adaptive proactive technique, presented in next sections, solves this issue by taking into account the device variability, as well, and enhancing the system lifetime.



## 4.5 Basis and Analysis of Variability-Aware Proactive Technique: A Static Non-Homogenous Approach

The proactive usage of existing available redundant units in the system gives the opportunity to each system column to go to recovery mode some time during its lifetime, since there exist some spares units in the system available to substitute the functional ones. As a consequence, when the column enters into recovery phase some part of the wear-out of its cells could be mitigated.

By adapting the recovery time of each element in the system in accordance with its time zero process variation, we can optimize the observed aging of the system elements, and as a consequence extend their lifetime even more. Therefore, the aged  $V_T$  values finally converge toward a common point meaning the optimal lifetime.

We consider again a system formed by five memory elements (columns), four active and one spare. Moreover, the duration of the recovery phase is static (non time-varying) but non-homogeneous. Thus, if the relative recovery ratio for a column  $i$  is given by  $D_i$  ( $0 \leq D_i \leq 1$ ), its active operating ratio is  $(1 - D_i)$ . Consequently, if the aging ratio (slope) of the device under continuous stress is given by  $M$ , then the aging ratio for this proactive reconfiguration technique is  $(1 - D_i) \times M$ , and the memory lifetime ( $T$ ) is obtained from Eq. 4.1:

$$(H - V_{Ti}) = (1 - D_i) \times M \times T \quad (4.1)$$

$$D_1 + D_2 + D_3 + D_4 + D_5 = 1 \quad (4.2)$$

$$D_i = \frac{H + (3 \times V_{Ti}) - \sum_{j=1, \neq i}^N V_{Tj}}{(5 \times H) - \sum_{j=1}^N V_{Tj}} \quad (4.3)$$

Eq. 4.1 shows the relation between lifetime ( $T$ ), the effective recovery

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

ratio ( $D_i$ ) and each column's threshold voltage  $V_{Ti}$  increase, required for a column to fail. According to Eq. 4.2 the sum of all the recovery phase ratios is 1 (100%); i.e. one column is always recovering.

Solving this equation system, the appropriate recovery ratios ( $D_i$ ) of each memory element are found in Eq. 4.3 making the  $V_T$  values to reach the common convergence point ( $H$ ) simultaneously, thus maximizing the system lifetime. In this way, Figure 4.4 exposes the results for the example with the aforementioned numerical  $V_T$  values and assuming an aging slope  $M = 10mV/year$ .

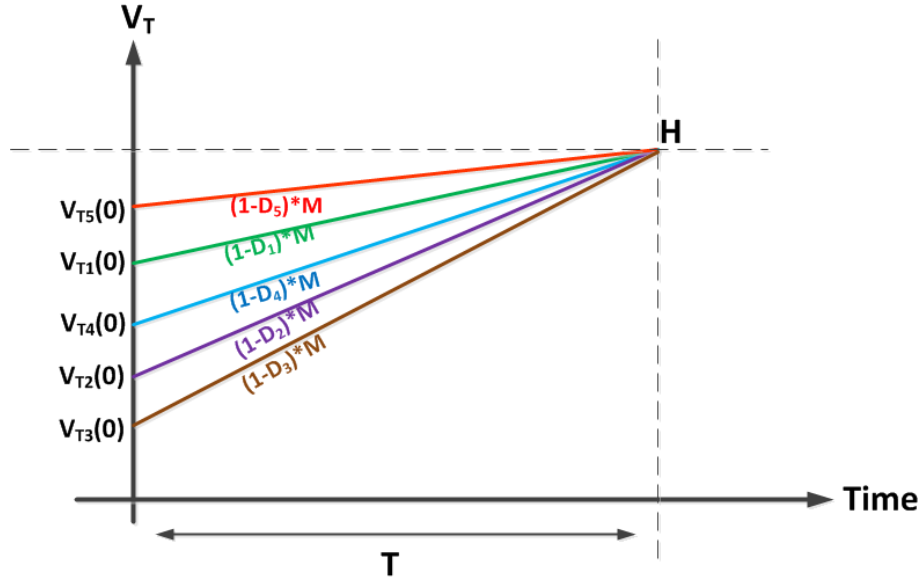


Figure 4.4: System behavior when an adaptive proactive technique is applied. The aging slopes change with adaptive recovery times, taking into account the process variability. Fresh  $V_T$  -values are shown on the vertical axis.

By solving Eq. 4.2 and 4.3, the resulting recovery time ratios in such proactive process variability-aware scheme of the example are given as follows:

$$D_1 = 0.28, D_2 = 0.12, D_3 = 0.02, D_4 = 0.20, D_5 = 0.38$$

In contrast to the homogeneous proactive technique, which uses equal recovery phases (IBM), this basis gives the following values:

## CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT IN SRAM ARRAYS

---

$$D_1 = D_2 = D_3 = D_4 = D_5 = 0.2$$

If we compare the obtained lifetimes of the three configurations, i.e. non-proactive Eq. 4.4, homogeneous proactive Eq. 4.5 and static variability-aware technique Eq. 4.6, we obtain that the system lifetime ( $T$ ) is given by:

$$T = \frac{H - V_{T5}}{M} = 7years \quad (4.4)$$

$$T = \frac{H - V_{T5}}{(1 - D_5) \times M} = 8.8years \quad (4.5)$$

$$T = \frac{H - V_{Ti}}{(1 - D_i) \times M} = 11.2years \quad (4.6)$$

These results reveal that the proposed technique can extend the memory system lifetime significantly ( $\sim 60\%$ ), when the activity of all the elements is well distributed based on their own  $V_T$  values and degradation status. Note that in this analytic analysis no wearout recovery was assumed. Next, we would consider a system containing overall  $N$  elements, which  $R$  of them are reserved as spare. Then, it is possible to obtain recovery period of each element by Eq. 4.7 as following:

$$D_i = \frac{(R \times H) + (N - R - 1) \times V_{Ti} - \sum_{j=1, \neq i}^N V_{Tj}}{(N \times H) - \sum_{j=1}^N V_{Tj}} \quad (4.7)$$

where  $D_i$  corresponds to recovery time of each element. Such proactive approach benefits from being simple and straightforward to utilize between memory columns as each memory column has two mode of active and recovery mode. Each memory column needs to go on the recovery time for a specific amount of time controlled by the CPU.

This technique allows us to adapt the aging of the memory columns in such a way that they all converge to a common point during their lifetime and it also removes the dependence of the system lifetime to the worst column's lifetime in the system. In next section, we extend this technique to a dynamic

adaptive basis, in order to take into account the inherent existing process variability and potential unbalanced aging behavior.

## **4.6 Dynamically Adaptive Extension to the Non-Homogeneous Proactive Technique**

Adaptive proactive reconfiguration is an improved version of the previous proactive reconfiguration [33] in which its utilization among memory columns results in a variability-aware utilization and dynamic balanced aging distribution, obtaining larger lifetime extensions throughout the memory columns.

First, we explain the overall approach flow of the proposed methodology, afterwards in more details the recovery time period calculation and characterization for each memory column. Next, we show the results of the proposed adaptive proactive approach in a memory system based on one spare unit, and finally we demonstrate the extension of the technique in this thesis to systems with more than one available spare unit.

### **4.6.1 Overall Approach Flow**

The proposed approach in this section is based on a non-homogeneous round robin sequence between all memory columns that also considers and self-adapts the process variation and BTI wear-out of SRAM cells in a time-varying basis. The utilization of spare units allows us to monitor the memories to determine the status of memory columns. It also permits to define different recovery times, which can be dynamically adapted to the respective  $V_T$  values. These different recovery time phase ratios will also compensate the differences in threshold voltages of SRAM cells in memory columns caused at the time zero by process variations and by aging during their lifetime.

Adaptive technique starts with a monitoring procedure in order to measure the SRAM cells  $V_T$  statuses (which are affected by process variation and BTI aging). The proposed monitoring approach can be implemented by monitoring the  $V_T$  gradual degradation in SRAM cells as to be presented in Chapter 5.

## CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT IN SRAM ARRAYS

---

This monitoring will evaluate the memory columns one-by-one and will measure the  $V_T$  value of each SRAM cell in each column. Then, each monitored column will be characterized by its highest  $V_T$  SRAM cell (the weakest cell in the column). These measured values which are regarding the device's time zero variation and BTI aging during the lifetime, determine the needed recovery time length ( $D_i$ ) for each memory column.

Note that the monitoring flow does not cause an idle time in the memory system, because the measurement can be performed while the corresponding column is in the recovery phase. So, Figure 4.5 depicts the procedure flow of the proposed adaptive technique. After the monitoring phase, the memory columns will be sorted from minimum to maximum (from the column with SRAM cell of highest  $V_T$  to the column with SRAM cell with lowest  $V_T$ ).

According to these values (maximum and minimum ones) specific dynamic ranges are defined, in an approach described in detail below. At the end, each memory column, with its weakest  $V_T$  value, will be classified into one of these ranges, where each one has a specific recovery period length.

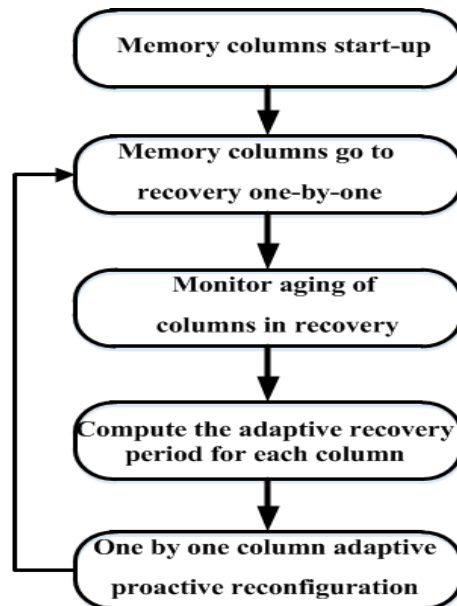


Figure 4.5: Adaptive proactive approach flow

### 4.6.2 Dynamic Recovery-Time Calculation

We use a dynamic approach to compute the required recovery time for each column. The proposed recovery period calculation approach is based on the range between the weakest and strongest SRAM cells  $V_T$  values, where min is the value of the minimum (best)  $V_T$  column, and max is the value of the maximum (worst)  $V_T$  column in the memory columns. It should somehow compensate the variation among the memory cells (process variation and aging) by a dynamic calculation of appropriate recovery period for each column.

Firstly, we consider a set of  $V_T$  ranges in which we want to classify the memory columns among them. Then, we calculate the  $\Delta V_T$ , which is the difference value between the best and worst column  $V_T$  values. Finally, the specific ranges are determined by the mentioned values. The columns are divided between these ranges in a manner that the columns with higher  $V_T$  values will have longer recovery times.

For instance, we have considered a case in Table 4.1 where the number of ranges is equal to four. Therefore, there exists four periods of recovery time, where each recovery period is calculated as mentioned above. As an example, the first recovery period is assigned to the group of  $V_T$  values that are between the min and the best  $V_T$  value plus a deviation, which is calculated as  $\frac{\Delta V_T}{4}$ . Note that the recovery periods are multiples of parameter named as  $T_R$ , which is the minimum, needed time for a complete BTI recovery, and it has been stated equal to  $10^4$  seconds [83].

Table 4.1: Round robin dynamic ranges

<b><math>V_T</math> ranges</b>	<b>Recovery time</b>
$min < V_T < min + (\frac{1}{4} \times \Delta V_T)$	$1 \times T_R$
$min + (\frac{1}{4} \times \Delta V_T) < V_T < min + (\frac{2}{4} \times \Delta V_T)$	$2 \times T_R$
$min + (\frac{2}{4} \times \Delta V_T) < V_T < min + (\frac{3}{4} \times \Delta V_T)$	$3 \times T_R$
$min + (\frac{3}{4} \times \Delta V_T) < V_T$	$4 \times T_R$

### 4.6.3 Single Spare Column Proactive Reconfiguration Case

The adaptive proactive reconfiguration puts the memory columns in the recovery mode, one by one since there is only one available spare unit. In each reconfiguration step, the spare column replaces a working column that goes into recovery mode and the column's data is copied in the spare column. When the memory column becomes active the copied data is written back in it before the next column reconfiguration step.

One reconfiguration cycle makes reference to when all the memory columns have experienced the recovery mode once. Each reconfiguration cycle can take up to several days, since the minimum recovery time that we consider is  $10^4$  seconds, in order to get a decent amount of  $V_T$ -recovery. As an example, we apply the adaptive proactive technique to memory columns with cells of their aforementioned  $V_T$  values of the example.

According to these values, for the couple of first reconfiguration cycles the column 5 will have the biggest recovery time. The reconfiguration will distribute the activity between columns in such a way, which at the end the values will become homogenous and uniform, and will converge to a single point. Table 4.2 presents an example of the flow for a single spare adaptive proactive reconfiguration between five columns (for instance  $C_5$  is considered as SP).

Table 4.2: Example of a round robin reconfiguration procedure

Time	Working Columns	Recovery Column
Memory Start-Up	$C_1, C_2, C_3, C_4$	SP
1st reconfiguration step	$SP, C_2, C_3, C_4$	$C_1$
2nd reconfiguration step	$C_1, SP, C_3, C_4$	$C_2$
3rd reconfiguration step	$C_1, C_2, SP, C_4$	$C_3$
4th reconfiguration step	$C_1, C_2, C_3, SP$	$C_4$
5th reconfiguration step	$C_1, C_2, C_3, C_4$	SP

In this context, Figure 4.6 presents the aging results in the proposed adap-

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

tive proactive technique among the 5 memory columns. It can be observed that the adaptive approach with  $R_f=30\%$  extend the memory columns lifetime in presence of process variability and BTI aging times. The obtained value is a 25% better than the IBM approach (dashed lines of Figure 4.3), 190 months in front of 150 and 200% better than the no-proactive approach 190 months in comparison with 84 months.

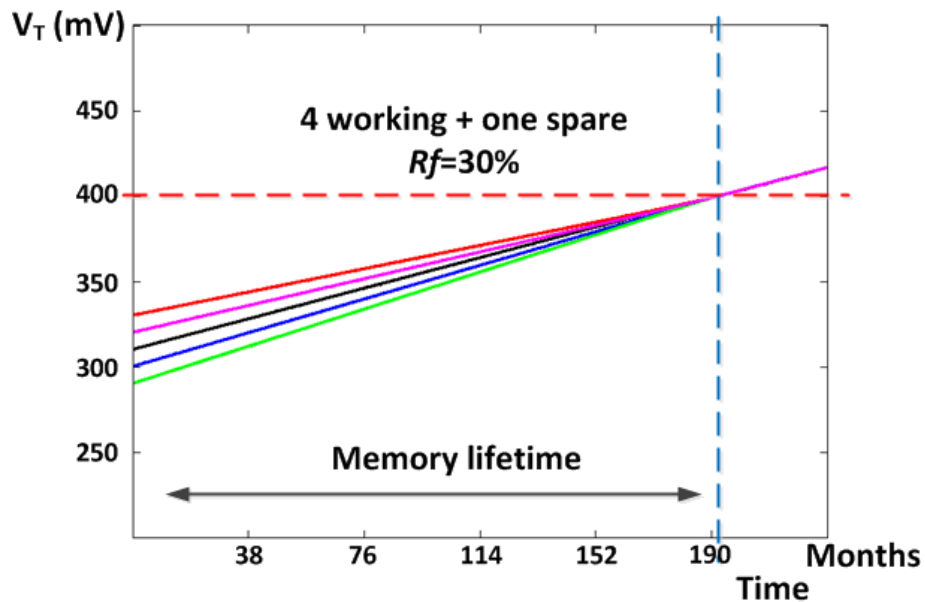


Figure 4.6: Adaptive proactive reconfiguration among 5 memory columns has resulted to uniform activity distribution and value convergence

Afterwards, by using Matlab simulations, we compare the lifetime of this thesis proactive proposal with a non-proactive reconfiguration scheme. To deal with it, we randomly generated fresh  $V_T$  values for a set of SRAM columns under normal distribution, with a given mean and standard deviation values (300mV and 30mV, respectively in the numerical examples).

The maximum acceptable  $V_T$  aging value ( $H$ ) before cell failure (when any dynamic parameter or the static noise margin, SNM, reach an unacceptable level) is assumed at 400mV. The wear-out recovery factor is assumed at two different levels: 30% and 50%, in order to include different technologies. Moreover, we have considered different number of active memory columns



## CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT IN SRAM ARRAYS

and variable recovery factor, by performing 1000 Monte-Carlo simulations in Matlab.

So then, Figure 4.7 , 4.8, 4.9 differ in number of operation columns and compare both proactive techniques' lifetimes in front of a non-proactive scenario. As expected, both proactive techniques extend the memory columns lifetime, although the proposed adaptive technique can enhance the lifetime more significantly. For this, in the following figures case 1 corresponds to the no-proactive approach where the system elements age continuously, case 2 corresponds to the homogenous recovery approach and case 3 corresponds to the adaptive proactive approach. Among these approaches the adaptive technique can extend the memory lifetime better and therefore would improve the system reliability.

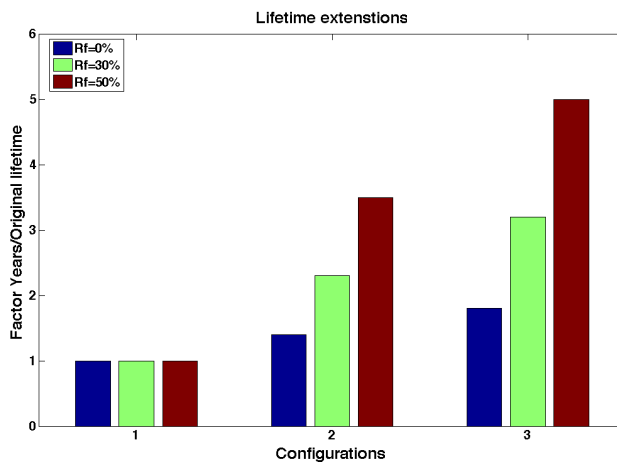


Figure 4.7: Lifetime improvement in a memory system with 4 operational columns and one spare, in different configurations of 1:Non-Proactive, 2:IBM-Proactive, 3:Adaptive-Proactive

These results show that as the number of active memory columns increases the lifetime extension in presence of only one spare reduces. This is due to the fact that having more spare units will allow the system to allocate more recovery periods to the operational units. In next section, we adapt this technique for the scenarios of having more than one spare memory column.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

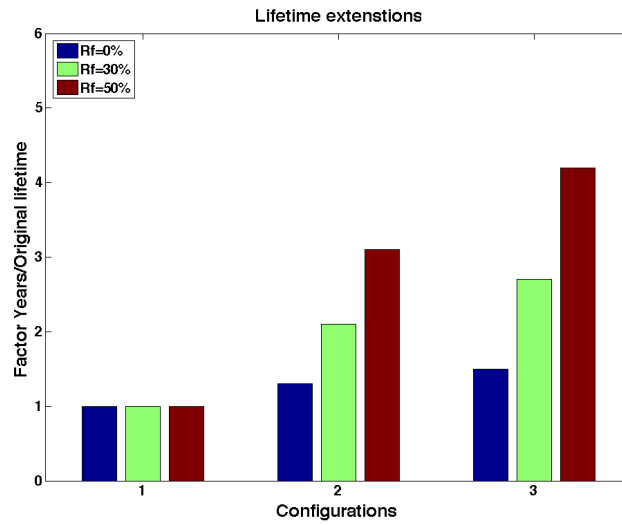


Figure 4.8: Lifetime improvement is a memory system with 8 operational columns and one spare, in different configurations of 1: Non-Proactive, 2:IBM-Proactive, 3:Adaptive-Proactive

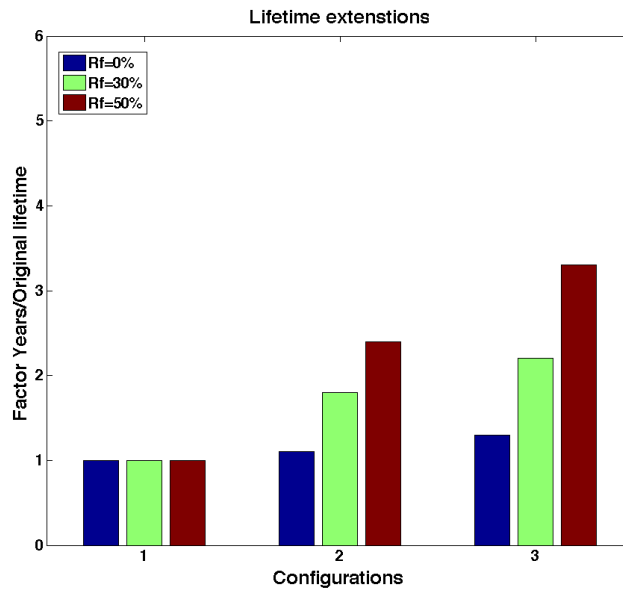


Figure 4.9: Lifetime improvement is a memory system with 16 operational columns and one spare, in different configurations of 1: Non-Proactive, 2:IBM-Proactive, 3:Adaptive-Proactive

#### 4.6.4 Multi Spare Columns Proactive Reconfiguration Case

When we extend the proposed strategy of reconfiguration scenarios, what contain more than one spare column in one memory column set we obtained improved results. Then, having more than one available spare column ( $R$ ) will allow to utilize them simultaneously in order to extend the system lifetime, and will lead to lifetime extensions even more than single spare adaptive proactive technique.

To adapt the presented proactive technique to multi-spare scenario, we perform an initial monitoring and classify the available memory columns in different classes. Each class contains memory columns, which have  $V_T$  values close to each other, and the number of columns in each class is equal to number of spare columns ( $R$ ). Then, similar to the single spare strategy in each reconfiguration step one class experiences the recovery mode.

The recovery time of each class is adapted with variability and degradation status of the column in the class with the highest  $V_T$  value. For this, Figure 4.10 points out the applied methodology for a case with 2 proactive spare columns and 8 operational columns as an example.

Then for instance, Figure 4.11 shows the convergence of  $V_T$  values in a system consisting of 8 functioning and two spare columns from adaptive algorithm implemented in Matlab. This system is divided by 5 classes (each class has two columns with their highest  $V_T$  values close to each other) and the wear-out recovery is assumed at 50%.

The adaptive proactive approach results in 3X lifetime extension and also balances the activity distribution between all SRAM cells. This demonstrates the benefits, in terms of system lifetime enhancement, of using adaptive recovery phase ratios during the columns' lifetime. The two convergence points are the effect of having two memory cells threshold voltages in each class, as the recovery period among the class members is equal; their difference in value will not converge.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

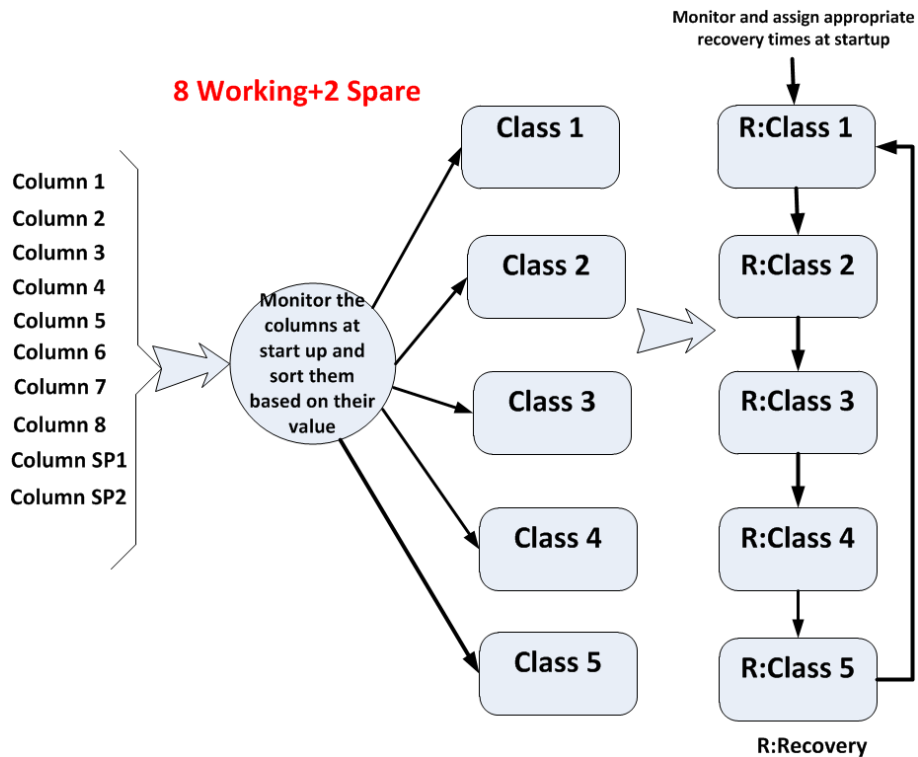


Figure 4.10: Multi-spare adaptive proactive reconfiguration

Table 4.3 depicts the results of 1000 Monte Carlo simulations showing the average lifetime without using adaptive technique and the lifetime improvements of memory columns in a multi spare adaptive technique when having a set of 8 and 16 operational columns, based on the presented approach.

We observe that as the number of operational columns increases the system lifetime could be enhanced by using higher number of proactive spare columns. For instance, when the recovery factor is 50% using one spare column can increase the lifetime about 3.3X, while using 2 spare columns with the same recovery factory enhance the lifetime about 4X, and finally having 4 spare columns can improve the lifetime of the column set around 5X.

CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT  
IN SRAM ARRAYS

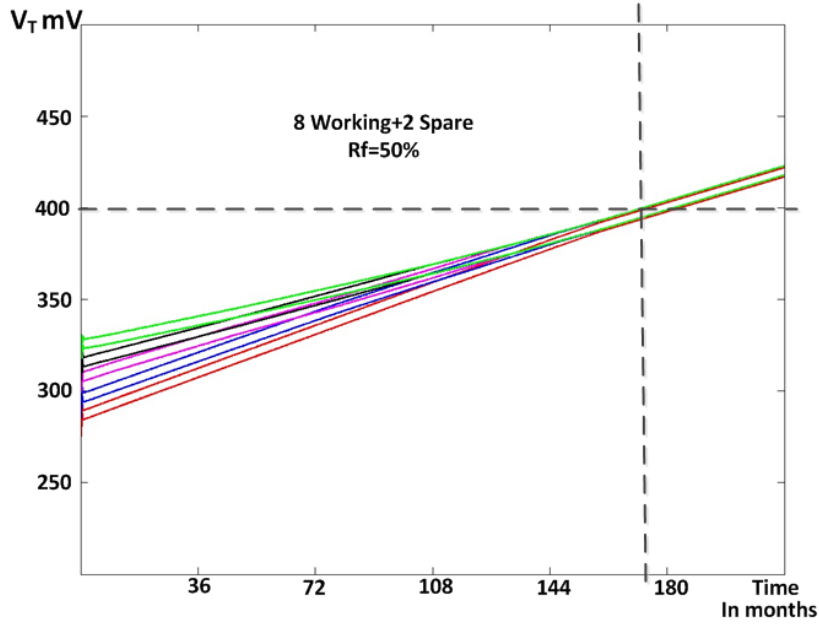


Figure 4.11:  $V_T$  aging slopes of columns in a adaptive proactive with multi-spare, 8 working and two proactive spare

Table 4.3: Memory lifetime extension in multi-spare adaptive reconfiguration in comparison with non-proactive

Working Units	Spare Units	Rf=0%	Rf=30%	Rf=50%
8	1	1.5X	2.7X	4.2X
	2	2X	3.2X	5X
16	1	1.3X	2.2X	3.3X
	2	1.5X	2.5X	4X
	4	1.9X	3.1X	5.1X

#### 4.6.5 Other Adaptive Proactive Approaches: Priority Sort, Based on Highest $V_T$

On the other hand, another adaptive approach can be based on measuring and monitoring all the devices threshold voltages of the memory cells in all columns (the operative and the redundant ones together) based on a

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

frequency (e.g. once per day) and to let the columns with memory cells that have the highest threshold voltages to experience recovery.

For example, if a memory structure has  $N$  working columns and  $R$  spare columns, the  $R$  columns with the highest threshold voltage will go to the recovery phase after every monitoring per day. Figure 4.12 shows the process flow for such an implementation.

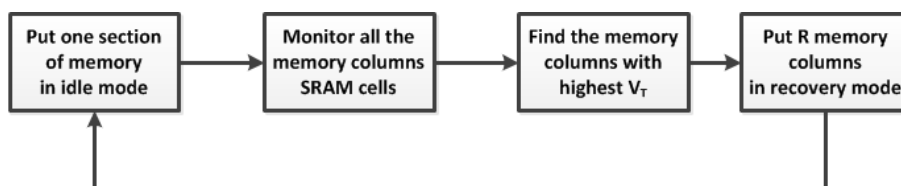


Figure 4.12: Priority sort proactive approach based on highest  $V_T$

This approach can also cause that the memory columns'  $V_T$  values to converge toward a common point value in the system lifetime. Additionally it results in larger lifetime extension of memory columns. For this, Figure 4.13 shows the convergence example of memory columns'  $V_T$  values (computed in Matlab with same modeling parameters as previous section). It is shown that time zero  $V_T$  values' variability, due to process variation, age along the lifetime, and finally they merge at a time point during the columns' lifetime. This technique is simple to implement in the hardware system, but it requires putting the memory columns of the memory section that we want to apply this kind of adaptive proactive mechanism in the idle mode in order to perform a monitoring phase and measure the threshold voltages. Such a technique needs special interrupts in the memory sections in order to monitor the memory columns and to define the  $R$  ones with highest  $V_T$  values in order to put them in the recovery mode. To show the benefits and compare both proposed techniques, as an example we have performed 1000 Monte Carlo simulations in Matlab, compared the two adaptive techniques and presented the results in Table 4.4.

## CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT IN SRAM ARRAYS

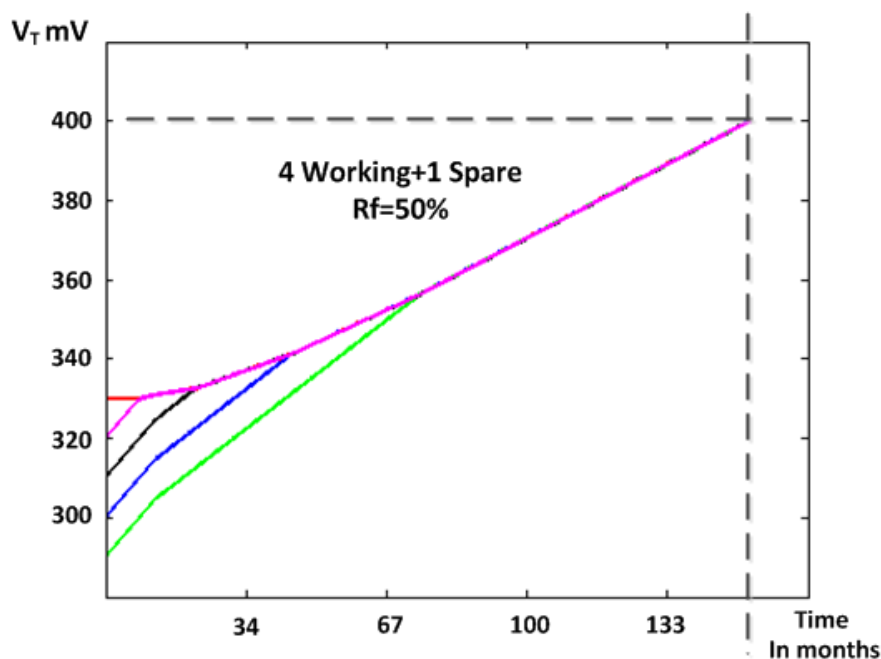


Figure 4.13:  $V_T$  aging slopes of columns in a priority sort proactive , the slope of aging of thershold voltages converges to a point along the lifetime

It depicts that between the two adaptive techniques, when the system does not benefit from any recovery mechanism, the priority-based approach has greater lifetime extension than the fully adaptive approach. However, as the recovery factor increases the fully adaptive approach can extend the lifetime more significantly.

Table 4.4: Relative lifetime extension obtained by the three proactive techniques over the non-proactive technique for a memory system based on five columns (4 operative + 1 spare)

	Rf=0%	Rf=30%	Rf=50%
Priority based technique	2.6X	2.8X	3X
Fully adaptive technique	1.8X	3.2X	5X

## 4.7 Impact of Adaptive Proactive Approach in SRAM Cell Design Parameters

In order to analyze the impact of  $V_T$  changes, due to BTI aging in nano-scale 6T SRAM cells, we establish the dependence of  $V_T$  shift to memory reliability metrics. The metrics analyzed in this thesis are the SNM and the  $V_{min}$ , since previous works have demonstrated that these are the mostly affected by the BTI aging [79], and other metrics such as the cell write margin might be negligibly affected by the BTI aging [79].

We also consider a static stress for the SRAM cells in which the cells store the same data for a long period of time. It has been illustrated that SNM, under static stress varies linearly with  $V_T$  shifts in FETs [79][90]. Therefore we have used a linear equation (Eq. 4.8) to relate the  $V_T$  shifts to the SNM, in thesis simulations:

$$\Delta SNM = -M \times \Delta V_{T,BTI} + C \quad (4.8)$$

To calculate the  $M$  and  $C$  parameters in Eq. 4.8 we have simulated and computed the SNM at 2 points; one at time 0 (non-stressed), and the other after a  $V_T$  shift of 50mV in the corresponding stressed N and P FETs of 6T SRAM. Finally, in order to analyze the impact of BTI aging in  $V_{min}$  drift of SRAM cells, we have considered a linear relationship between the memory SNM and the  $V_{min}$  [90][91][92]. Therefore, the SNM drop due to device aging, results in linear increase of  $V_{min}$ .

Figure 4.14 shows the convergence of SNM values, computed in Matlab, in a system consisting of 4 functioning and 1 spare column. This system is based on presence of one supply voltage ( $V_{min_{active}}$ ) and the wear-out recovery (Rf) is assumed at 30%. The adaptive proactive approach balances the activity distribution between all SRAM cells, thus demonstrating the benefits of using adaptive recovery phase ratios during the columns' lifetime. It can extend SRAM columns' lifetime in comparison with no-adaptive proactive memories.

Also while the no proactive memories need to increase the  $V_{min}$  of the memory to keep it functional, the proposed adaptive approach can reduce



## CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT IN SRAM ARRAYS

this  $V_{min}$  drift (because of slower SNM drop) along a specific period of time (5 years in this analysis).

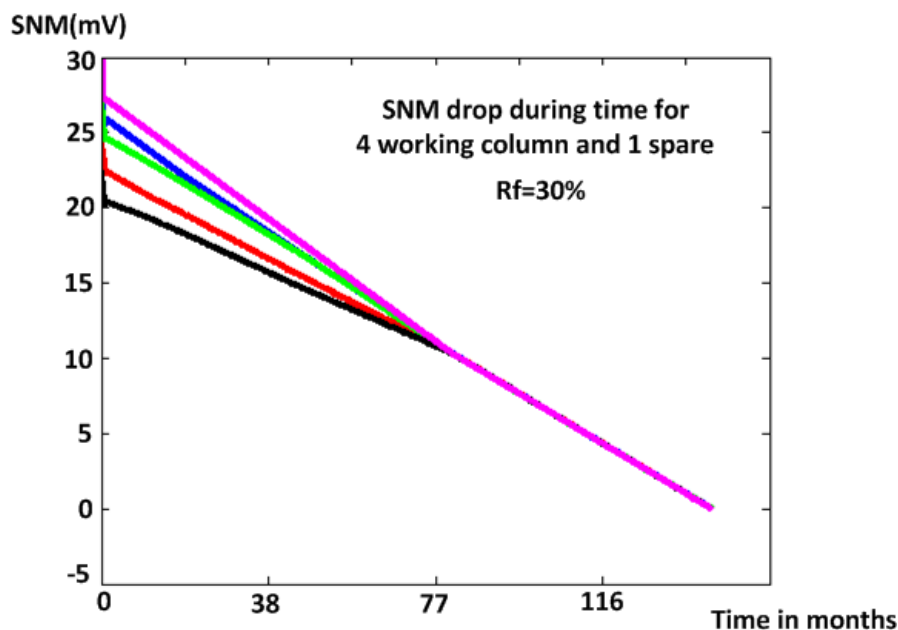


Figure 4.14: Degradation behavior of weakest SRAM  $SNM_s$  of memory columns due to BTI aging by utilizing adaptive proactive reconfiguration

To show the benefits of this technique, we have performed 1000 Monte Carlo simulations in Matlab to compare the adaptive proactive approach versus the non-adaptive approach with various number of operational and spare memory columns and presented the results in Table 4.5. The adaptive technique depicts a relevant reduction of the  $V_{min}$  drift in SRAM by slowing down the  $V_T$  increase among its devices. Now that this thesis has explored the benefits of utilizing an adaptive proactive approach, the next goal would be to implement such architecture to realize this technique. Regarding this, in the next section a hardware implementation approach is described for an example of 1kB SRAM array with the proactive reconfiguration methodology.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

Table 4.5: Relative lifetime extension and  $V_{min}$  reduction drift obtained by the proactive over the non-proactive technique for a memory based on different number of spare and operational columns

Configuration	$V_{min}$ drift reduction
4+1	30%
8+2	27%
16+1	15%
16+2	25%

### 4.8 Architecture for the Proactive Technique and Overhead Evaluation

In this section, an architecture is proposed for a SRAM cache memory system with an adaptive proactive compatibility. For instance, it is assumed that the SRAM capacity is 1kB, and contains 8 spare columns. In this section the main framework of the corresponding architecture is introduced. Next Chapter 5, will present the details of monitoring procedure and circuits to monitor the process variability and aging of SRAM cells in the columns. This section also analyzes the proposed reconfiguration control scheme to manage the whole adaptive proactive approach. Finally, the overall area overhead of the adaptive approach in 1kB SRAM is evaluated.

#### 4.8.1 Architecture

The architecture for a SRAM cache memory with proactive compatibility is obtained by adding complementary circuits to the original memory architecture. The considered 1kB SRAM array example consists of 128 columns divided into 8 groups of 16 columns and each memory column contains 64 6T-SRAM cells.

It is considered that the 1kB memory contains 8 spare columns and each one of the spare columns belongs to one set of the 16 columns. Therefore, the



## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

We use a single circuit to monitor all the columns of the memory; this avoids the impact of relative variability in the monitor circuit itself and results in a low overhead. Note that larger memory blocks can also be constructed by the architecture shown in Figure 4.15, where a defined partition of columns can share the same monitoring circuit.

Figure 4.16 depicts the structure of one memory column (column0) in the thesis proposed approach. The column (BL and BLB) is connected to a monitoring circuit by the  $TM_{L0}$  and  $TM_{R0}$  transistors, and they are activated independently in the monitoring phase. Two transmission gates and control signals ( $TS_x, TS_{Tx}$ ) isolate the column memory cells from the undesired word-line during the normal memory operation and the monitoring phase.

In this sense, the monitoring process does not interfere with the normal operation of the other memory columns (active ones). The SRAM cells in the columns will be monitored and their aging status will be recorded inside the register file. This aging information would be utilized by the controller to dynamically adapt the recovery time between the memory columns, it will also allow the appropriate column to experience recovery through the 2-1 multiplexers. Two control signals and two transmission gates at Figure 4.17 (a pair of one NMOS and one PMOS transistor) are proposed, so that node S can swing from  $V_{DD}$  to Gnd in the normal and monitoring mode of the memory operation. For instance Figure 4.18 shows the datagram of waveforms at the important nodes during the normal and monitoring phase of SRAM column.

CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT  
IN SRAM ARRAYS

---

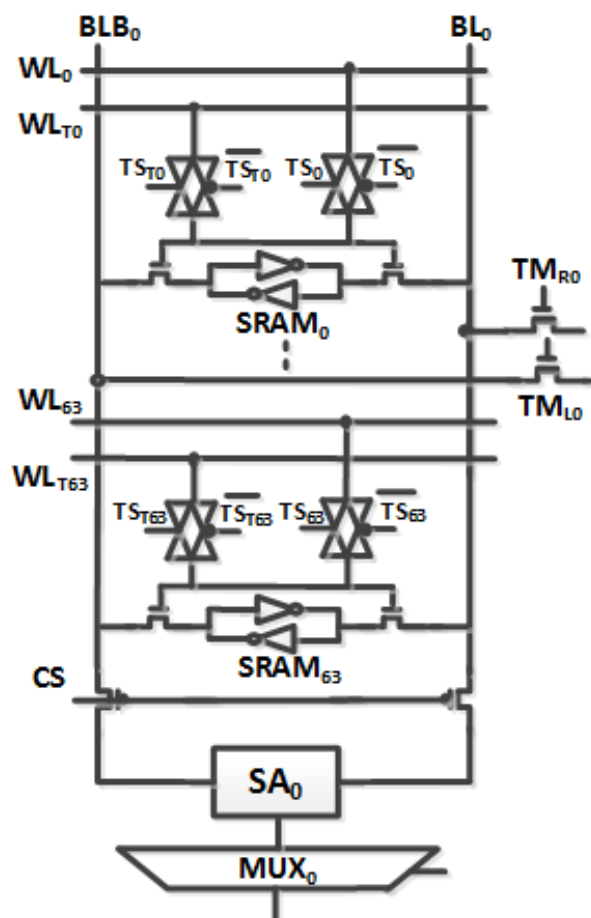


Figure 4.16: Structure of one SRAM column in the adaptive proactive re-configuration

Table 4.6 summarizes the waveforms shown in Figure 4.18. It presents the status of the control signals during the normal and monitoring phase of SRAM column. Note that each phase is also divided to two parts, where one is the access state and another one is the hold.

We have included these transmission gate transistors in area calculation in the part of total word-line pass transistors (Table 4.7). We have also considered this fact in the overall area calculation of the proactive technique (Table 4.8).

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

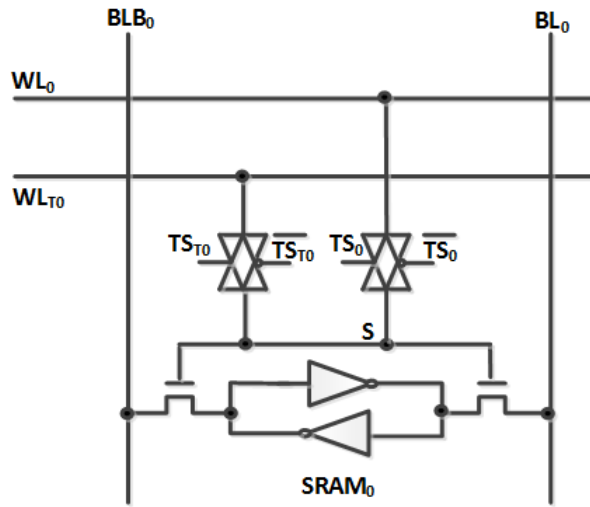


Figure 4.17: SRAM cell connection to Word-lines

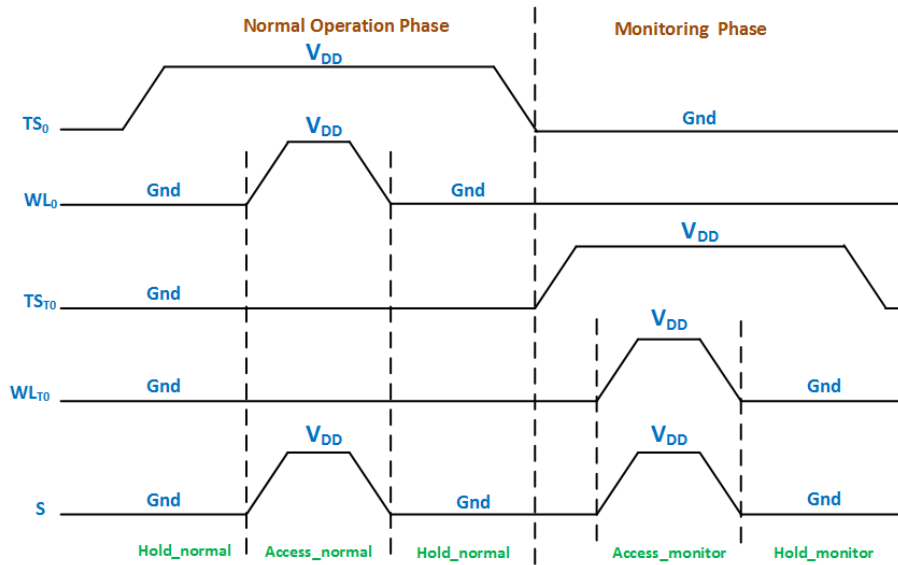


Figure 4.18: The node voltages Waveform

All the shown proactive techniques except for common recovery approach (IBM) need a monitoring mechanism and in addition a control scheme to be applied among the memory columns cells. The monitoring mechanism is left to be explained in detail at Chapter 5 of this thesis, but the reconfiguration control scheme and the area overhead evaluation are described in following.

## CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT IN SRAM ARRAYS

Table 4.6: Modified Node Voltages at each stage operation

	Access Normal	Hold Normal	Access Monitoring	Hold Monitoring
<b>TS<sub>0</sub></b>	$V_{DD}$	$V_{DD}$	$GND$	$GND$
<b>WL<sub>0</sub></b>	$V_{DD}$	$GND$	$X$	$X$
<b>TS<sub>T0</sub></b>	$GND$	$GND$	$V_{DD}$	$V_{DD}$
<b>WL<sub>T0</sub></b>	$X$	$X$	$V_{DD}$	$GND$
<b>S</b>	$V_{DD}$	$GND$	$V_{DD}$	$GND$

### 4.8.2 Control Unit

The reconfiguration control unit manages the monitoring circuits and the procedure of recovery calculation. Additionally, it controls the switching of the columns between active and recovery mode and outputs the correct column by controlling the multiplexers. Figure 4.19 shows the block diagram of the proactive reconfiguration control unit.

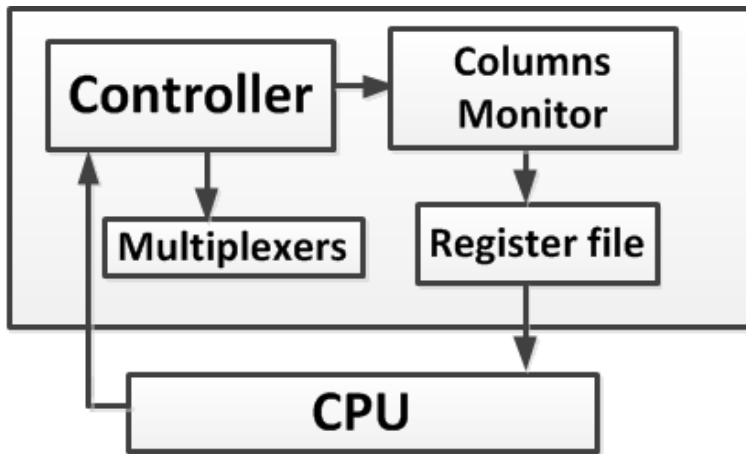


Figure 4.19: Reconfiguration control scheme

The CPU itself is included in the control saving area; it reads the digital values of BTI aging of the SRAM cells in each memory column, which is recorded in a register file. Then it calculates the appropriate and needed recovery period for each column, taking into account their worst SRAM cell

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

$V_T$  -value in terms of time zero process variation and BTI aging.

Each column and the corresponding spare column in that set are connected to a 2-1 multiplexer (Figure 4.15). The controller is a built-in state machine that controls the switching of the columns between active and recovery mode and outputs the correct column through appropriately selecting the control signals of the multiplexers.

### 4.8.3 Area Overhead Evaluation Results

Area overhead is always a relevant metric for industry, for this we have evaluated the presented implementation in terms of silicon area overhead through area estimation. First, Table 4.7 lists the components used in monitoring circuit of the SRAM architecture and the corresponding estimated area for each one in 45nm technology node.

According to calculated estimation the main area overhead is regarding to the word-line pass transistors. Note that, the comparator area overhead is extracted from the integrated comparator circuit in [93] which can also be utilized in the monitoring circuits of this work.

Next, to obtain the overhead related with the digital units (such as the register file, counters and the state machine), their corresponding structure is coded in VHDL and then synthesized with RTL compiler toward CMOS 45nm LP (low power) technology library [94]. So as conclusion of the total area overhead, Table 4.8 states the area overhead estimation, the area of monitoring circuit implementation and the overall proactive reconfiguration monitoring design. Consider that the overall extra circuit implementation requires a silicon area of  $670\mu m^2$ , which is around 12% of the 1kB 45nm SRAM silicon area in [94] ( $6000\mu m^2$ ).

Note that, the other reconfiguring units used to implement the adaptive methodology, such as the controller and the multiplexers, can be merged with the already existing and utilized units in the reactive reconfiguration of memories (conventional reconfiguration and redundancy techniques in memories).



## CHAPTER 4. PROACTIVE RECONFIGURATION CONCEPT IN SRAM ARRAYS

Table 4.7: Components Area Size Used in Monitoring Circuits

Components	Area in 45nm( $\mu m^2$ )
Total TM transistors	$17 \times 2 \times 8 \times (0.018) = 5$
Total word-line pass transistors	$17 \times 64 \times 8 \times 2 \times 2 \times (0.004) = 140$
Current mirrors	$2 \times 2 \times (0.05) = 0.2$
Resistors	10
Digital current sources	$2 \times 2 \times 20 \times (0.05) = 4$
Comparator	100 [93]
Reserved for routing overhead	60
<b>Total</b>	<b>320</b>

Table 4.8: Adaptive Proactive Monitoring Design Area in SRAM Array

Unit	Area ( $\mu m^2$ )
1kB 45nm 6T SRAM	6000 [94]
Controller, register files and counters	350
Monitoring circuit	320
Overall monitoring units for the proactive design	670( $\sim 12\%$ )

The implemented methodology of this thesis in the SRAM arrays will slightly impact the memory cache performance. This reconfiguration mechanism is done in such a way that in each column reconfiguration step, the CPU copies the working column's data that goes into recovery mode in the spare column and this copied data is written back in the column before the next column reconfiguration step, with no impact on normal operation.

One complete proactive reconfiguration of all the memory columns can take up to couple of days and the frequency of reconfiguration process among the columns is very low, which allows the copying process to have enough time in order to let the two columns contain the same data. Therefore, the small performance loss would be only at the switching time of a column to another, and the monitoring process of the recovery column can be a DC measurement.

## 4.9 Summary and Conclusions

In this Chapter, a new basis to improve the existing proactive mechanism in SRAM arrays has been proposed. This is based on managing the recovery allocation between SRAM columns by considering both the time-zero variability and time-dependent variability (BTI aging). The efficiency of proposed adaptive technique is first presented graphically, and then Monte-Carlo simulations have shown its benefit. It is demonstrated that allowing the SRAM columns to experience dynamic recovery periods during their lifetime can extend their working lifetime up to 5X, and therefore, a significant improvement of their reliability could be obtained.

We have shown the concept of adaptive proactive reconfiguration can be extended to more than one spare column among the SRAM columns. This fact would enhance the lifetime extension in existence of BTI aging and time-zero variability. Also an alternative adaptive approach based on priority recovery is presented which can also extend the memory lifetime and its results are compared with the main proactive technique of this thesis.

To implement the adaptive proactive strategy in SRAM array, this thesis proposes an architecture, which can drive the reconfiguration inside the memory system. Such architecture benefits from monitoring algorithm and circuits, and the needed units to control and manage the reconfiguration. The overall overhead of this implementation is compared with a conventional SRAM system, and it is calculated to be 12%. In overall, the adaptive proactive technique which is proposed in this Chapter can be an efficient approach for mitigating aging effects in SRAM cells, to be utilized in future embedded memory structures.

All of the results Chapter has been published in Conference papers of VLSI Test Symposium (VTS 2012) [88], International Conference Mixed Design of Integrated Circuits and Systems (MIXDES 2012) [95], Design Automation Test Europe (DATE 2013) [96], International Symposium on Quality Electronic Design (ISQED 2014) [97] and Journal paper of IEEE Transaction on VLSI (TVLSI 2015) [98].

# 5

## Monitoring Procedure in SRAM Arrays with Reconfiguration Mechanisms

### 5.1 Introduction

Traditionally, the aging effects in transistors were measured one by one and manually through off-line techniques, such as utilizing microscopic electrodes, to probe an individual device inside the silicon wafer [99]. However, this approach has become inefficient with increase of number of transistors in a chip with millions of transistors. Therefore, better monitoring techniques have emerged for both logic and memory circuits to measure subtle effects of aging.

Monitoring aging in logic is mainly based on measuring timing violations for example through Ring Oscillator (RO) sensors [100]. In such approaches the phase difference between two ring oscillators, one stressed in actual operation and the other one in no stress is checked to predict the aging profile inside the chip [101].

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

In addition to logical units inside the microprocessor, the aging in memory cells can be monitored by current based techniques [102][103][104][105][106]. The cells with more aging, will have unbalanced *SNM* and weaker devices, therefore an aging profile can be created. The precision on aging sensor depends on the current sensor, and this is an important factor in applying repairing steps appropriately.

In respect to existing monitoring approaches inside SRAM arrays this Chapter of the thesis aims for design of a novel on-chip aging sensor in order to reflect the aging profile of the SRAM cells. This proposal will monitor aging in the SRAM transistors and provide the information for the global proactive approach to reconfigure the SRAM configuration appropriately.

Adaptive Proactive Reconfiguration was previously introduced in Chapter 4 as a technique to improve the reliability in SRAM arrays in the presence of BTI aging mechanism and process variations. In this sense, one of the key principles in this approach is the ability to monitor the system elements in function of accumulated variability and aging effect, and later to use the corresponding monitoring information to improve the overall system reliability.

Additionally, another important aspect that makes monitors a critical element in the design of proactive schemes in memories is the workload variation, which impacts on the temperature and supply voltage inside the chip [107]. This would impose an un-balanced aging inside the chip, and would make it difficult to distinguish the specific circuits that experience the performance degradation.

Nevertheless, an on-chip aging monitoring circuit can provide valuable measurement of performance degradation. This information can be used to apply mitigating techniques in those specific parts of the chip to extend the system lifetime. These mitigating techniques can be Adaptive Body Bias (ABB) [17], Dynamic Voltage Scaling (DVS) [17], or utilizing the existing redundancy proactively to balance the workload and aging among all units.

As an example, consider a SRAM memory constructed with numbers of cell arrays, in which one of them is aged more in respect to others. In this case to operate the SRAM reliably, one solution is to run the memory with lower capacity ignoring the array, which may cause failure. But, if moni-

## CHAPTER 5. MONITORING PROCEDURE IN SRAM ARRAYS WITH RECONFIGURATION MECHANISMS

toring circuits exist, then specific mitigations or less workload or intelligent reconfiguration can be applied to the respective section with higher aging.

### 5.2 Related Work

Regarding existing monitoring approaches, this section introduces some of them to measure the degradation of SRAM cells.

#### 5.2.1 IDDQ Testing

The first approach is based on monitoring NBTI in SRAM arrays using the Iddq-based sensing [102]. By tracking and measuring the standby leakage current in SRAM arrays it is possible to predict, within a reasonable accuracy, the lifetime when the SRAM cell is subjected to NBTI degradation.

Note that BTI aging results in  $V_T$  increase in transistors, and consequently the leakage current decreases, as well. So then, simulations exhibit that reduction in leakage current closely follows the same linear behavior in a PMOS transistor (Figure 5.1). Therefore, one can predict the memory lifetime by monitoring its leakage.

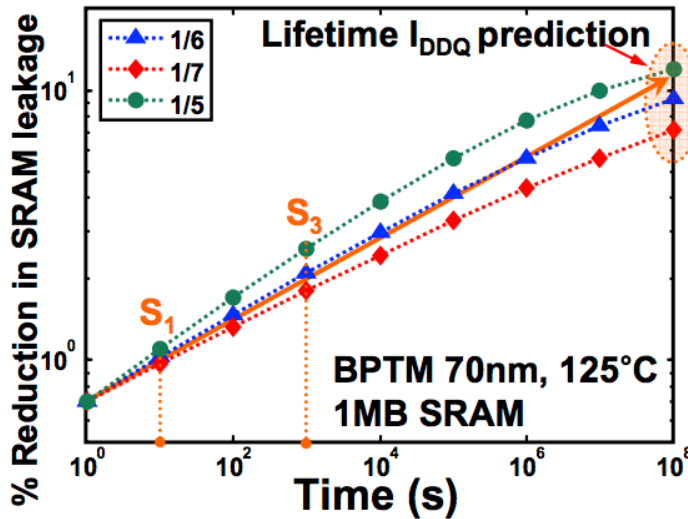


Figure 5.1: Impact of time exponent change in Iddq degradation trend [102]

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

To constantly monitor the SRAM  $IDDQ$  during the memory lifetime a built in  $IDDQ$  monitoring circuitry is shown in Figure 5.2. In the  $IDDQ$  test mode the MB device is off and the SRAM array  $IDDQ$  current is mirrored to MN and an analog voltage signal  $V_{out}$  reflects changes in SRAM  $IDDQ$ , while in the normal memory operation mode the MB transistor is turned on by  $V_{BYPASS}$  signal to avoid performance loss due to the monitoring circuit.

This approach can only measure the overall leakage current of the whole SRAM array, and by this it can define total amount of BTI aging. However, it does not consider the existing variability among the SRAM cells because it is based on column-by-column measurement. It cannot give information about the reliability status of each individual SRAM cell and cannot diagnose the worst or near to fail SRAM cell in the array.

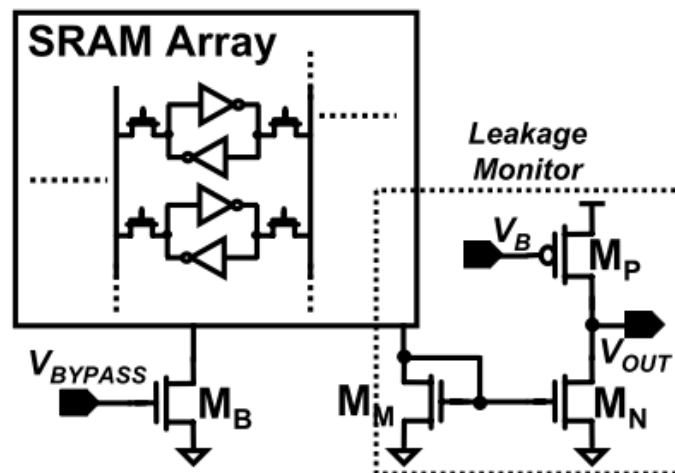


Figure 5.2:  $IDDQ$  Monitor circuit for SRAM array [102]

### 5.2.2 NBTI on chip monitoring using BIST

In [103] authors propose an on chip reliability monitoring design that can monitor the degradation of both PMOS devices in each individual SRAM cell of cache array. Their approach does not affect the normal device operation and can predict the failure in a specific cell. However, their design needs  $2 \times N$  ( $N$ =number of word size) times replication of the Design for Testa-

## CHAPTER 5. MONITORING PROCEDURE IN SRAM ARRAYS WITH RECONFIGURATION MECHANISMS

bility (DFT) circuitry which can increase the area overhead of the proposed design. Additionally, the on chip monitoring circuitry can be sensitive to the process variation and aging which may affect the degradation monitoring of the SRAM cells.

An important factor of this proposal is that this current based technique can monitor the NBTI degradation in each one of the SRAM cells without affecting the performance of the cell during the normal operation. This can be due to the SRAM cell writability condition that ensures the access transistors are stronger than the pull up transistors and the output nodes of the two SRAM inverters can be brought close to zero. In this sense, Figure 5.3 depicts a cell in this state when both bitlines are forced to ground. This puts the cell in a metastable state where the inputs and outputs of the inverter are forced to zero.

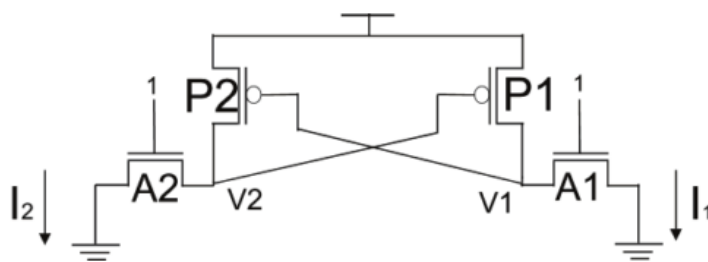


Figure 5.3: SRAM cell state for NBTI testability [103]

It is demonstrated that at this state the output currents  $I_1$  and  $I_2$  are insensitive to variations in the NMOS devices of the cell, and are appropriate indicator of the PMOS transistor states. Therefore, by sampling these currents it is possible to monitor the NBTI degradation in each SRAM cell.

### 5.2.3 Few other approaches to monitor BTI aging in SRAM Array

In addition to the two approaches introduced above there exist other techniques to measure aging in SRAM arrays where a few of them are presented here. One approach to track the degradation among the SRAM cells is stated

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

in [104]. It utilizes a 6T SRAM cell as an embedded monitoring sensor in the SRAM array to monitor the aging of a subset of the cells. The drawback of this method is the necessity of having many sensors for a complete memory array monitor and also it is not a precise degradation monitoring technique of a specific SRAM cell.

Another proposed on-chip aging sensor is based on connecting a sensing scheme to each SRAM column and periodically performing off-line tests, by monitoring the write operations on the SRAM cells [105]. Figure 5.4 shows a global schematic diagram of this approach where the On Chip Aging Sensor (OCAS) can measure the aging in the SRAM cells.

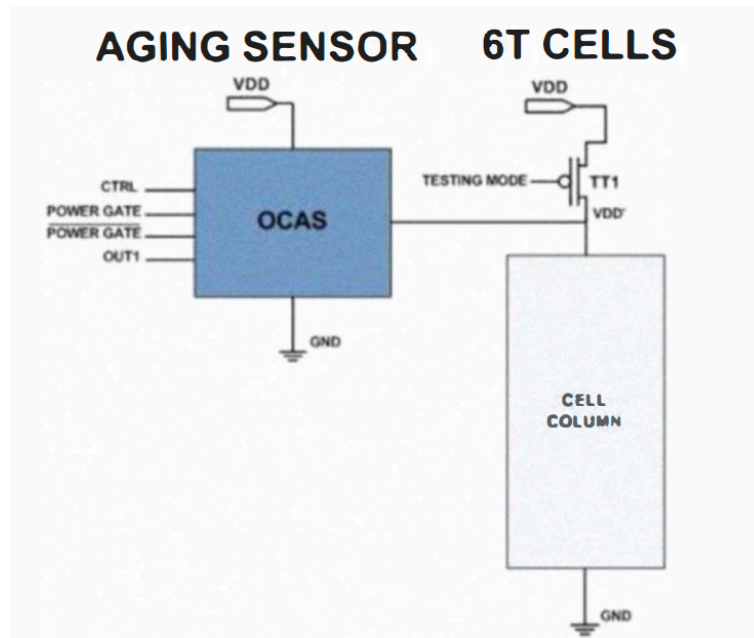


Figure 5.4: SRAM cell aging sensor [105]

Although this approach can monitor aging efficiently, but the added monitoring circuits and the measurement complex algorithm makes it difficult to be implemented in SRAM chips.

Finally, recently a new proposal [106] uses an in-situ technique to sense the threshold voltage of SRAM PMOS transistors directly. This work has similarities to the proposed technique in this Chapter since it also measures the aging by connecting two transistors to the SRAM bitline. By applying





## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

NBTI and PBTI) wear-out and variability status of individual SRAM cells in each memory column in a DC manner.

This proposed technique monitors the SRAM cells degradation in a column-by-column sequence. This strategy has no effect on the normal memory operation, since it is applied when the specified column is in recovery mode, and it is disconnected during the normal operation of the column.

The proposed monitoring circuit is based on two current mirrors, which are connected to the memory column bit-lines. They track the current passing by each SRAM transistor and since the current value depends on the device status, in this way we can also analyze the process variation among the all SRAMs' transistors.

Figure 5.6 shows a scheme of the proposed monitoring circuit for a SRAM cell in an specific case. The monitoring process for each column starts when the column goes to recovery mode and it requires two steps. The first one is to write a logic value '1' in all the SRAM storage nodes. Then, a counter enables each word-line (WLT) one-by-one in order to measure the aging and process variability value of the pull-down transistors (the right NMOS, NR) and pull-up transistors (the left PMOS, PL) in SRAM cells of specific column.

As an example, to monitor the NR transistor aging in the first column (column0) in Figure 5.6, the controller enables the switches TMR0 and T4 (the switch that selects the appropriate current mirror in respect to the N or PMOS), selects the input2 (the voltage value is generated by current mirror and the resistor) from Mux1, and input2 (voltage generated by digital current source and the resistor) from Mux2. Then, it enables switches TML0 and T1 and selects input1 from Mux1 and Mux2 to monitor the aging in PL. Next, a '0' is written to all the SRAM storage nodes in the column, and again the counter turns on each word-line (WLT) one-by-one and this time the monitoring circuit tracks the degradation in (NL) and (PR) transistors of SRAM cells.

**CHAPTER 5. MONITORING PROCEDURE IN SRAM ARRAYS WITH RECONFIGURATION MECHANISMS**

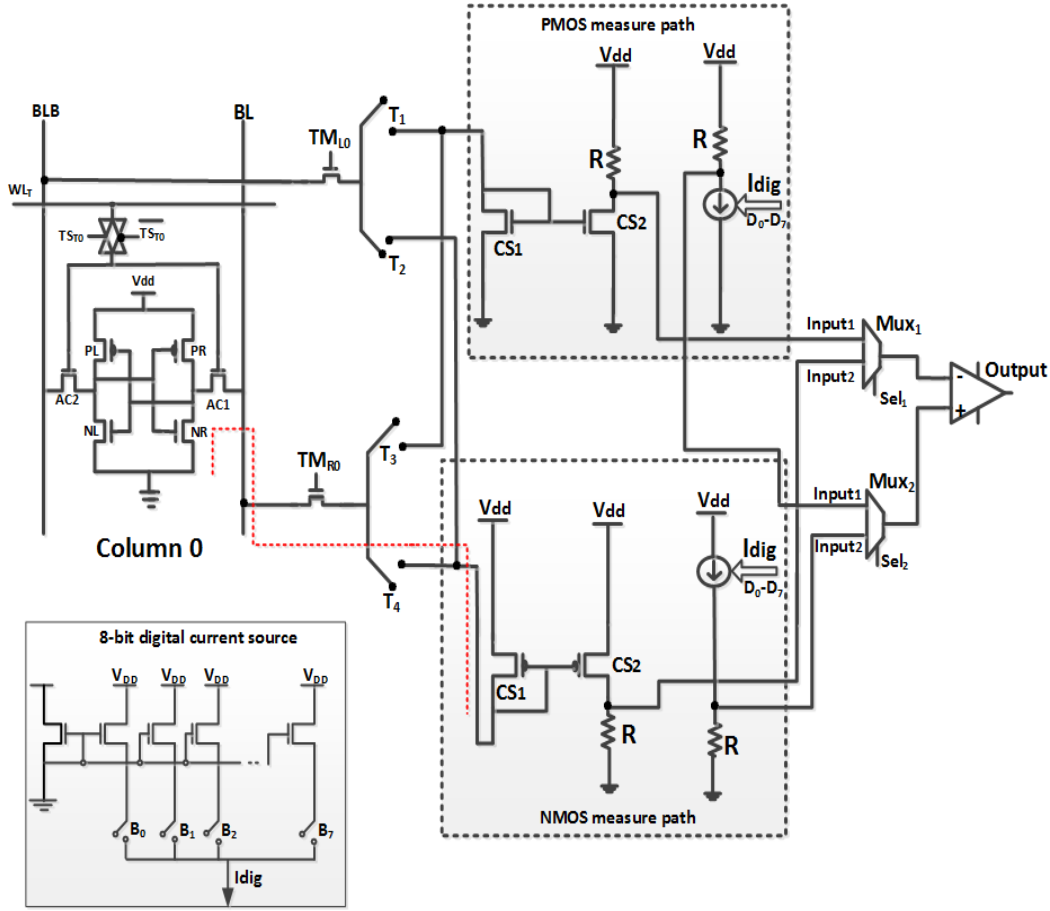


Figure 5.6: Process variation and aging sensing scheme in column 0, the inset shows the digital current source implemented by the current mirrors

It is worth noting that, to avoid short channel effects and assure a good match of mirrored current, the devices used to measure the aging performance, i.e. TM and the current mirror transistors, are designed as long and wide channel devices. Note that since the monitoring circuit is applied to all the columns to order them based on their aging value, the possible mismatch or deviation does not have significance in the columns ordering.

So then, first, we have designed and simulated the presented monitoring technique by using 45nm Predictive Technology Model (PTM) transistor [108] in HSPICE [109]. As a matter of example, we have assumed a logic value '1' is written in the SRAM storage node, and we monitor the aging in NR transistor of the SRAM cell.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

The dotted red line inserted in Figure 5.6 illustrates the measurement path to monitor the performance of this specific SRAM transistor. The  $I_{NR}$  current is mirrored in the current mirror, and when the current digital current source ( $I_{dig}$ ) gets equal to  $I_{CS2}$  the comparator output changes its state and the current value is recorded in the register file.

The  $I_{dig}$  is shown in the inset of Figure 5.6 and is implemented by a current mirror with elemental sources, and has a resolution of 8 bits. The measured current flows through the path made by SRAM pull down NR, access transistor (AC1) and the monitoring switch (TMR0) device.

The TMR0 transistor is designed with larger size than the SRAM cell transistors, and only switches during the cell's monitoring mode, so it should be slightly affected by aging and process variations. The access transistor (AC1) is also slightly affected by aging because it only switches when the cell is accessed from the decoder to read or write into the cell. Therefore, the important transistor to monitor aging in the path is the SRAM transistor (NR).

Furthermore, when the current flows in this branch at the monitoring phase, the AC1 and TMR0 transistors are at their linear region while the (NR) transistor is in saturation region, so the measured current value will highly depend on the NR transistor current, and not on the access and TMR0 transistors.

In this sense, it will be simulated how the variability affects the monitoring circuit performance. Hence, first, Figure 5.7 depicts the monitoring analysis for the current measurement where the aging (PBTI) is emulated by a voltage source connected to the gate of the transistor, and measure the aging in NR.

In the monitoring phase, the access transistor (AC1 considered with variability) and the TMR0 transistor are biased on (operating in linear region), and the NR is biased with the storage node voltage. Then, Figure 5.7 also shows the transistors in their operating region. Afterwards, Figure 5.8 shows the degradation in a NMOS device as an example, and the relation between their  $V_T$ -shift and current weakening, after 1000 Monte-Carlo simulation at each aging point.

It is observed that as the NMOS is stressed the  $V_T$  starts to shift down

## CHAPTER 5. MONITORING PROCEDURE IN SRAM ARRAYS WITH RECONFIGURATION MECHANISMS

from its nominal value, and the device current reduces in accordance with it. Also it demonstrates that the current swing due to aging in NR is large enough and slightly affected by the process variation in access transistor, therefore we can order the columns correctly in respect to their aging.

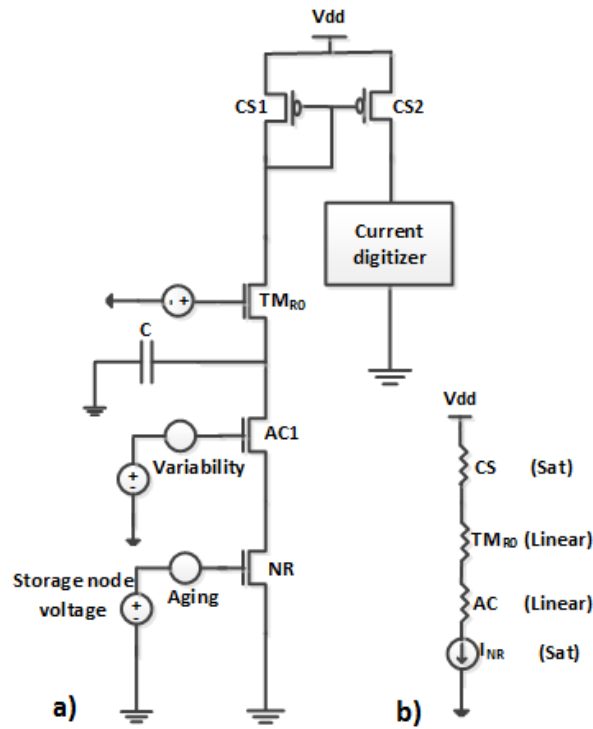


Figure 5.7: a) Monitoring simulation scheme, b) Transistors in their operating region

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

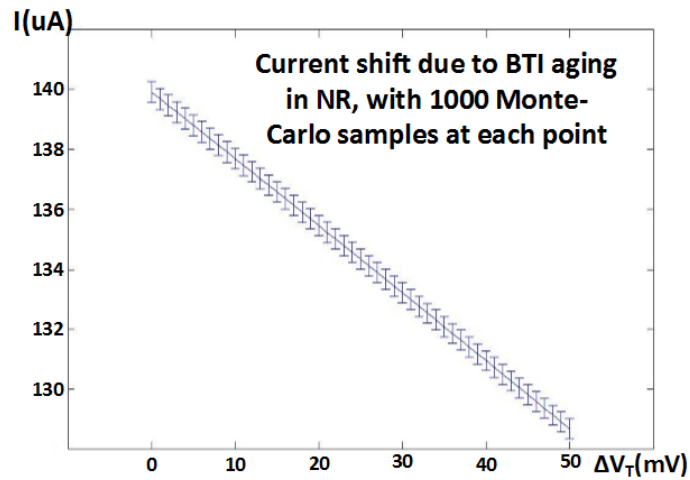


Figure 5.8: Current decrease during  $V_T$  shift, representing aging of a NMOS

Finally, Figure 5.9 presents the current value in the branch, after a 1000 sample Monte-Carlo analysis and considering variability in AC1 and fixed aging and variability in NR transistors. It is expected that, the NR transistor which is in the saturation region, will direct the current in this path and the access transistor would have much less impact on the current.

This claim is validated in the presented simulations. The data from HSPICE simulations have been extracted to Matlab and plotted, showing that the access transistor variability has a small impact on the current and the greatest distribution of current value is around the nominal NR current value of 140uA. Therefore, it is the NR transistor that has the biggest impact on the current in the measurement path.

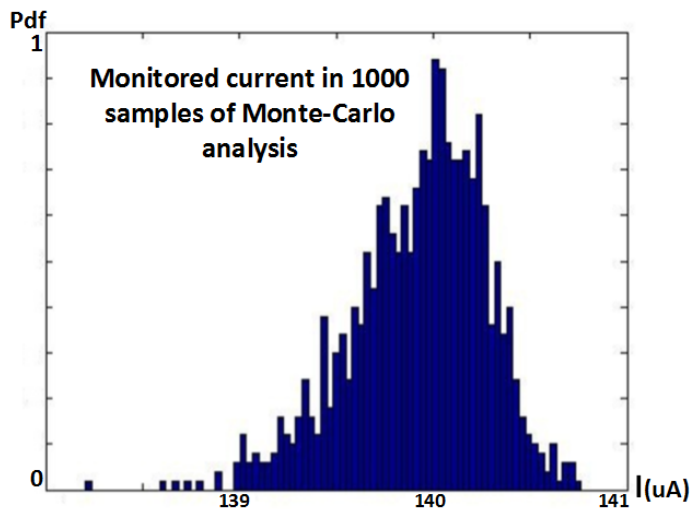


Figure 5.9: Monte-Carlo simulation considering variability in access transistor

## 5.4 Experimentation

To demonstrate the feasibility of the proposed technique, in this Chapter, we have designed and fabricated an integrated chip in CMOS 350nm technology node of AMS manufacturing kit.

Note that, the purpose of the manufactured chip is not to compare the aging and variability between modern CMOS with 350nm technology nodes, but to experimentally analyze the presented technique as a verified aging ( $V_T$  shift) monitoring technique for SRAM PMOS transistors ( $SiO_2$ ).

PMOS transistors are chosen for aging monitoring, because at this technology node NMOS transistors are not built with high-K as a gate dielectric material, and consequently their PBTI aging is negligible. Moreover, we should mention that, to emulate the  $V_T$  shift in the 350nm PMOS transistors, we modify the device body bias, since the process variation and BTI aging is not significant in the 350nm technology node.

Figure 5.10 shows the schematic view of the fabricated sensor to monitor the aging and process variation status of the PMOS transistors of each SRAM cell. Both PMOS 1 and 2 are the transistors that are mostly affected by

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

the aging (NBTI) [79]. Therefore, they are the devices, which undergo the measurement procedure.

We have fabricated this sensor in a 350nm CMOS technology node, with a nominal voltage ( $V_{DD}$ ) of 3.3V. In order to emulate the aging and variability effects, we use the body bias shift mechanism in each PMOS transistor. Note that the same sensor circuitry utilized for aging measurement in PMOS 1 can be used to monitor the aging in PMOS 2 through multiplexing.

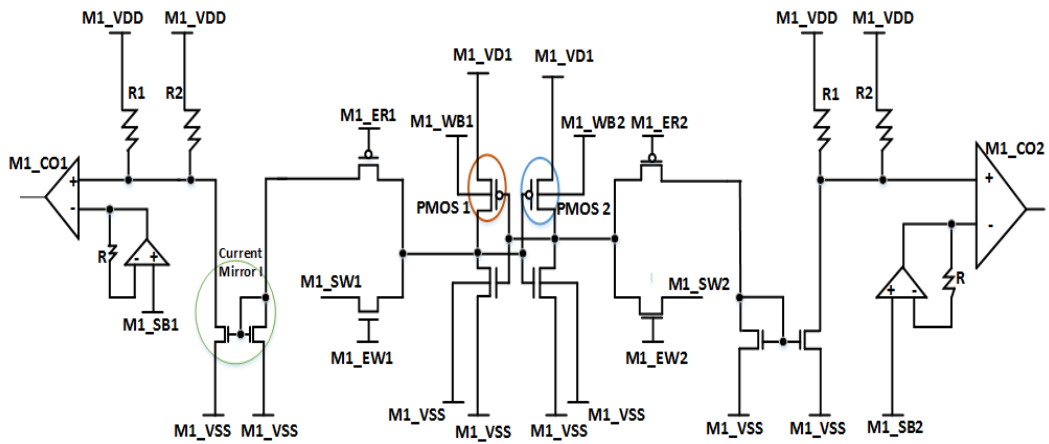


Figure 5.10: Schematic view of the fabricated on-chip aging sensor

In this context, Figure 5.11a depicts some of the monitoring datagrams of the built chip simulation in Cadence [110], where we monitor the degradation in one of the SRAM PMOS transistors. In this monitoring sequence of datagrams, first we write a '1' inside the SRAM storage node and later we sense (read) the current through the respective PMOS, when this current passes a known threshold, the comparator output changes and we can measure the aging status of the specified FET.



## CHAPTER 5. MONITORING PROCEDURE IN SRAM ARRAYS WITH RECONFIGURATION MECHANISMS

Figure 5.11b depicts the impact of changing the PMOS body bias in the monitoring technique. We have performed a parametric analysis in Cadence and increased the PMOS body bias step by step. As depicted, this causes that the comparator output to change state at different time points regarding the  $V_T$  shift in PMOS and its current.

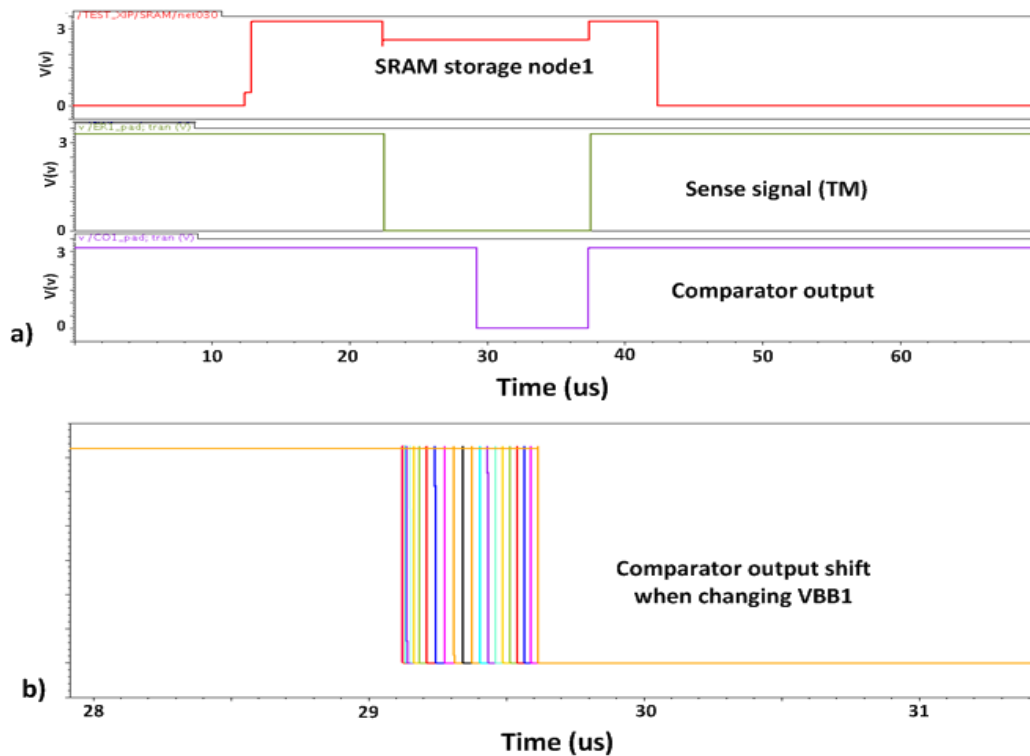


Figure 5.11: a) Monitoring sequence of writing the value one in one of SRAM storage nodes and sensing the aging in the corresponding PMOS, b) Comparator output shift when changing body bias voltage of PMOS PL

In the following Figure 5.12 depicts the floor plan of fabricated chip under the microscope. Note that the chip has been packaged in DIP and then soldered to the PCB for the measurements. Figure 5.13 shows the experimentation setup schematic and in the lab.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

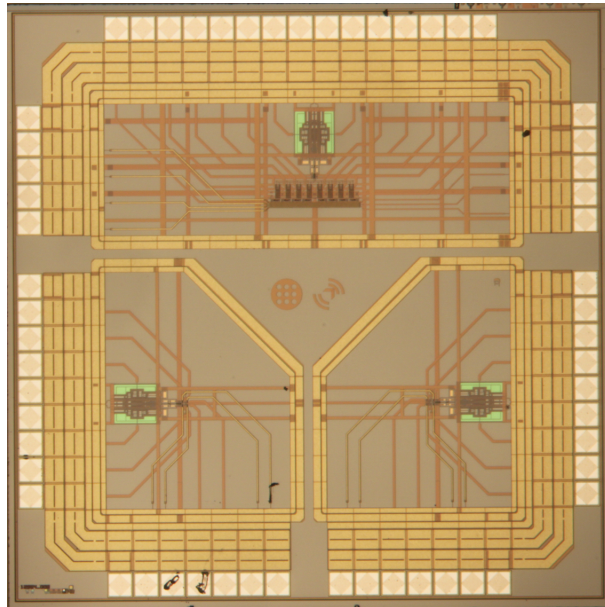


Figure 5.12: The fabricated chip floor plan

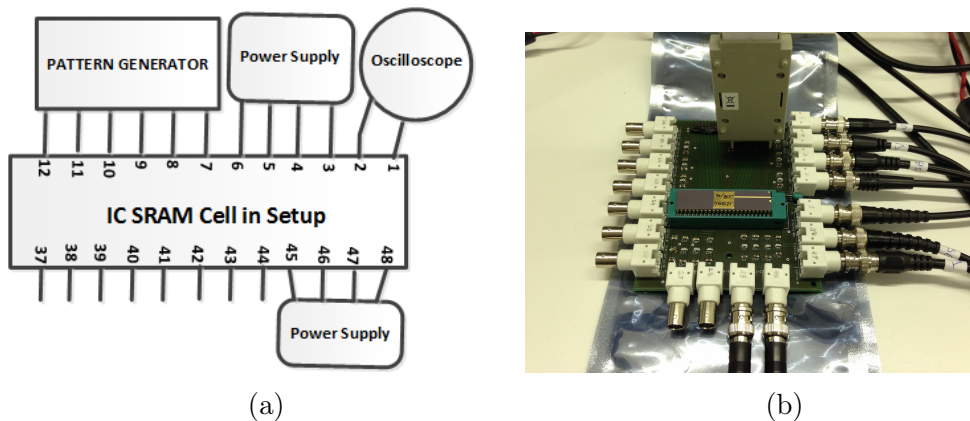


Figure 5.13: The chip measurement setup in the lab, a) Schematic for IC connections, b) The PCB connections in the lab

To measure the aging status in the left PMOS (PMOS 1) of the SRAM cell in the lab; first, we set the PMOS body bias at a given value, emulating a device aging. Then, we write '1' in the SRAM cell by applying the appropriate signals with a pattern generator, shown in Table 5.1.

Afterwards, we perform a current sensing process. In this step by applying the corresponding input signals with pattern generator, the PMOS 1

## CHAPTER 5. MONITORING PROCEDURE IN SRAM ARRAYS WITH RECONFIGURATION MECHANISMS

---

current is mirrored through current mirror I (Figure 5.10), and converted to a voltage to be compared with an external voltage ( $M1_{SB1}$ ) in the comparator ( $M1_{CO1}$ ). Once the comparator output changes its status (from high to low) the external voltage represents the current in the PMOS 1 of the SRAM.

Figure 5.14a presents the obtained results (10 chip prototype samples) of the sensor measurements when PMOS 1 suffers from aging and variability. The measurement procedure is continued in the next step by writing a '0' value inside the SRAM cell. This brings the PMOS 2 to active region and applying the appropriate signals it would be possible to make the measurement.

Note that, the measurement step can be performed, similarly to the PMOS 1, by applying the appropriate signals presented in Table 5.2. So, Figure 5.14b shows the result of measurement for PMOS 2 and for 10 chip prototypes. The impact of aging is shown with degradation (reduction) of the current with the sweep of body bias voltage.

Furthermore, the impact of process variation is observed in the difference of each measurement line for every chip prototype. So then, we can observe that the proposed sensing technique can measure the impact of emulated aging and variability ( $\sim 15\%$  variation in measured PMOS current) in the SRAM cell precisely. Therefore, with this information obtained from the on-chip aging SRAM sensor, the proactive reconfiguration technique can properly manage the working load of each memory column to enhance the overall system lifetime.

**RELIABILITY-AWARE MEMORY DESIGN USING  
ADVANCED RECONFIGURATION MECHANISMS**

---

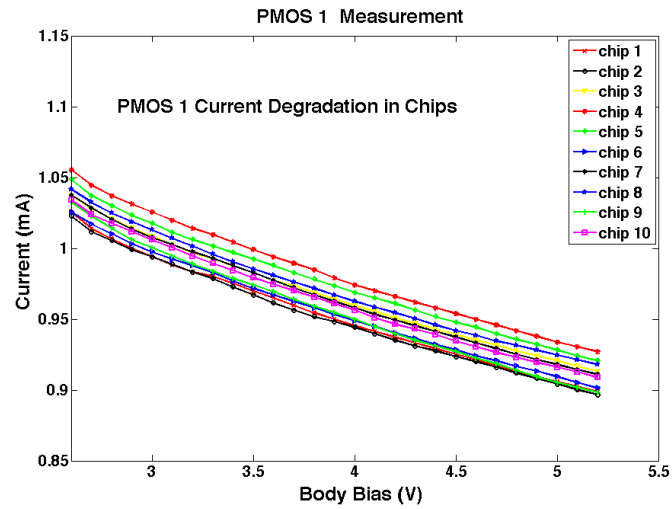
Table 5.1: Write and sense signals to monitor the aging in PMOS 1

Signal	Value in Write Mode	Value in Sense Mode
$M1_{SW1}$	$V_{DD}$	GND
$M1_{EW1}$	$V_{DD}$	GND
$M1_{ER1}$	$V_{DD}$	GND
$M1_{SW2}$	GND	GND
$M1_{EW2}$	$V_{DD}$	GND
$M1_{ER2}$	$V_{DD}$	$V_{DD}$

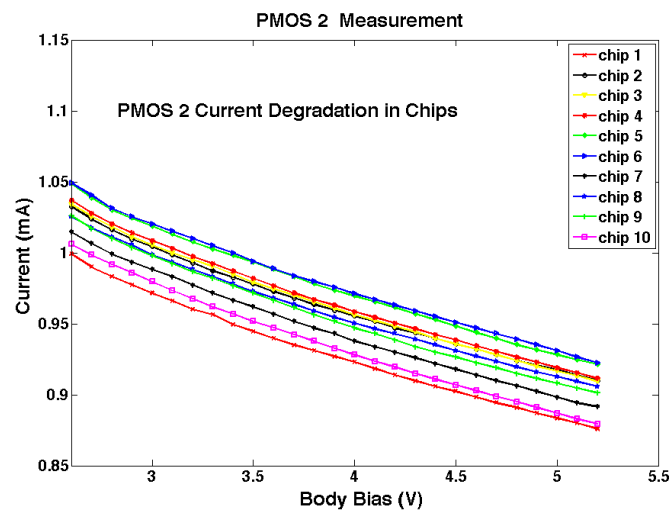
Table 5.2: Write and sense signals to monitor the aging in PMOS 2

Signal	Value in Write Mode	Value in Sense Mode
$M1_{SW1}$	GND	GND
$M1_{EW1}$	$V_{DD}$	GND
$M1_{ER1}$	$V_{DD}$	$V_{DD}$
$M1_{SW2}$	$V_{DD}$	GND
$M1_{EW2}$	$V_{DD}$	GND
$M1_{ER2}$	$V_{DD}$	GND

## CHAPTER 5. MONITORING PROCEDURE IN SRAM ARRAYS WITH RECONFIGURATION MECHANISMS



(a)



(b)

Figure 5.14: Aging and variability measurement in a) PMOS 1 b) PMOS 2

## 5.5 Summary and Conclusions

In this Chapter we have designed and implemented novel circuitry proposal to monitor the aging and process variation status of SRAM cells by using a built-in sensor. The proposed monitoring approach is based on measuring the current in critical transistors of SRAM cell and it does not influence the

## **RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS**

---

normal dynamic operation of the memory.

In comparison with previous monitoring techniques the proposed approach in this thesis considers the simultaneous effects of process variability and BTI aging in SRAM cells. Circuit simulation results have shown the applicability of this sensing approach inside the memory.

Next, we have designed and built/fabricated an integrated circuit with the purpose of monitoring technique in a 350nm CMOS technology. This chip includes the complete circuits to measure the aging and process variation in one SRAM cell as a matter of proof of concept.

The presented results from the chip depict the relation between SRAM reliability and the measured current of its transistors. Therefore the experimental measurements from the chip in the lab verify the feasibility and efficiency of this monitoring approach and show the applicability of this novel monitoring technique in SRAM cells.

All of the results Chapter has been published in Conference papers of Design Automation Test Europe (DATE 2013) [96], International Symposium on Quality Electronic Design (ISQED 2014) [97] and Journal paper of IEEE Transaction on VLSI (TVLSI 2015) [98].

# 6

## Process Variability and Reliability Concerns in Memristive Memories

### 6.1 Introduction

As it was explained in the introduction Chapter, memristor devices have appeared as one of the promising candidates to be implemented at ultra small technology nodes. In this sense, the following Chapter will review three of the more important models and behavioral mechanisms for memristor devices presented nowadays in the literature. Moreover, significant existing reliability concerns in those devices will be also introduced such as process variability and endurance degradation. These reliability factors are mainly originated from nano-scale dimension mechanisms in the memristor devices.

For instance, while process variability could cause variation in the nominal high and low resistance value, endurance degradation will impose dynamic variation to the resistance values due to aging mechanisms. Finally, this Chapter will analyze the impact of these reliability concerns in the normal operation of memory cell, and evaluate the read and write instability due to variability and endurance degradation.

## 6.2 Memristor Models and Mechanisms

Among the different behavioral mechanisms and development of the memristor models, in the literature there are a few ones that have attracted more considerations. So then, this thesis considers the three main switching mechanisms to model their behavior, as: ionic drift [40], pickett [111] and conductive filamentary model [112] [113].

Note that there also exist other models for memristors in the literature, which can be found in other research works [114][115]. In this section we briefly introduce the three main mechanisms and models considered in this thesis.

### 6.2.1 Ionic Drift Model (HP model)

In this model the behavioral mechanism of memristor relies on linear dopant drift of oxygen vacancies by an external voltage bias applied across the device. In such a way, employing a positive voltage to the doped region will rebuff the oxygen vacancies and extend the doped area; meanwhile a negative voltage would act in reverse.

These changes of dopant region would cause a switch in the resistance value in respect to rate of change of dopant drifts, where the value of Low Resistance State (LRS) corresponds to the lowest resistance value because of high dopant concentration and High Resistance State (HRS) to highest resistance value due to non-existence of dopants.

This proposal was published by HP [40], where they presented the first physical model of a memristor. Their device was constructed by two metal layers (e.g. platinum Pt) and a thin film metal oxide of thickness  $D$  (e.g.  $TiO_2$ ) sandwiched between them. The memristor ionic drift model is based on the following equations of Eq. 6.1, 6.2, 6.3:

$$x = \frac{w}{d}, 0 < x < 1 \quad (6.1)$$

$$R_{mem} = LRS \times (x) + HRS \times (1 - x) \quad (6.2)$$



## CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES

---

$$\frac{dx}{dt} = \left(\mu \times \frac{LRS}{D^2}\right) \times i(t) \quad (6.3)$$

where  $R_{mem}$  is the total memristance value,  $x$  is the normalized state variable (obtained by dividing the original state variable  $w$  to the oxide thickness  $d$ ) and, therefore its rate of change depends on memristor thickness and the current passing through it, LRS and the dopant mobility ( $\mu$ ).

The above equations can result in pinched hysteresis loops of memristors in the device simulation. However, the model described by them can get stuck at the boundaries of the state variable  $x$  (0 and 1), and also it does not emulate the natural non-linear behavior of dopant drifts manifested at thin film edges. Therefore, to solve these problems a window function ( $f(x)$ ) is defined and multiplied into Eq. 6.2 as follow in Eq. 6.4:

$$\frac{dx}{dt} = \left(\mu \times \frac{LRS}{D^2}\right) \times i(t) \times f(x) \quad (6.4)$$

The window functions in [116] can overcome both issues, nevertheless it makes the model predictivity dependent on it. In this context, Figure 6.1 shows the memristor ionic drift switching concept.

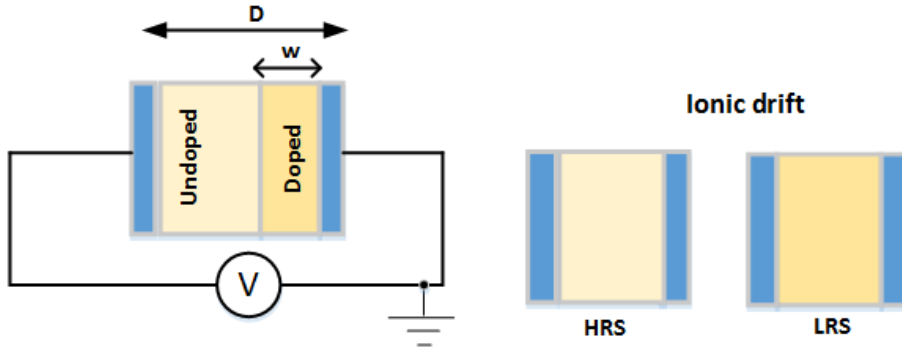


Figure 6.1: Memristor switching mechanism based on ionic drift model, applying voltage across the memristor can change the size of doped region ( $w$ ) therefore changing the resistance state of the memristor

The ionic drift model was one of the first models to describe the memristor behavior, but it could not explain all mechanisms in such devices like the electron tunneling and filament creation. Therefore, more models were

investigated and proposed to describe those effects.

### **6.2.2 The Pickett Model**

The second mechanism that could describe the memristor dynamic switching is originated from a physics-based model [111]. In this model, the drift diffusion of vacancies in the oxide film is explained by change of  $w$  (effective distance of the tunneling gap), because of tunneling distance modulation under an applied voltage or current.

This model is characterized by the tunneling current equations, and it is very sensitive to the changes of input signal, however it features the non-linear dynamics of memristors. This type of memristor switching mechanism is defined by the following equation of Eq. 6.5:

$$I = w^n \beta \sinh(\alpha V) + \chi(\exp(\Upsilon V) - 1) \quad (6.5)$$

Above, the first term represents the electron tunneling behavior through a thin residual barrier, where  $\alpha$  and  $\beta$  are fitting parameters and  $w$  is state variable of the memristor normalized between 0 and 1. Moreover, the second term of Eq. 6.5 contains the  $I - V$  representation for rectifier and  $\chi, \Upsilon$  are fitting parameters.

### **6.2.3 The Conductive Filamentary Switching Model (CF)**

Finally, the third memristive operation kinetic is based on conductive filamentary (CF) switching. This model is very similar to the dielectric breakdown effect, where the formation and disruption of the CF results in memristor switching to low (LRS) and high resistance values (HRS).

Initially the forming process constructs a filament between top and bottom electrodes without connecting them to each other; in this state the memristor is in its HRS mode. Applying a positive voltage at the top electrode moves the ions through the insulating layer and the ion formation between the metal contacts extends the CF and reduces the resistance toward the LRS value, depending on the CF width.

## CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES

In order to switch back the memristor to its HRS mode, a voltage with opposite polarity would reverse the ion migration process and will rupture the CF toward the HRS mode. Note that, the explained conductive filament process also depends on material used for fabrication of memristor [1].

For instance, in some type of ReRAM devices the filament is constructed by metallic ions while in some other type it is the oxygen ions, which form the filament and manage the conductive filamentary behavior. The main equation describing the CF growth and dissolution is presented and formulated in Eq. 6.6 [112]:

$$\frac{d\phi}{dt} = Ae^{-\frac{EA_0 - aqv}{kT_0(1 + \frac{V^2}{8T_0\rho k_{th}})}} \quad (6.6)$$

Where  $\phi$  is the CF diameter,  $EA_0$  is the energy barrier for ion hopping,  $V$  is applied voltage to memristor,  $a$  is barrier lowering coefficient,  $q$  is elementary charge,  $k$  is the Boltzman constant,  $T_0$  is the room temperature,  $\rho$  is electrical resistivity,  $k_{th}$  is the thermal conductivity and  $A$  is a constant. Figure 6.2 shows the conductive filamentary switching model in memristive devices.

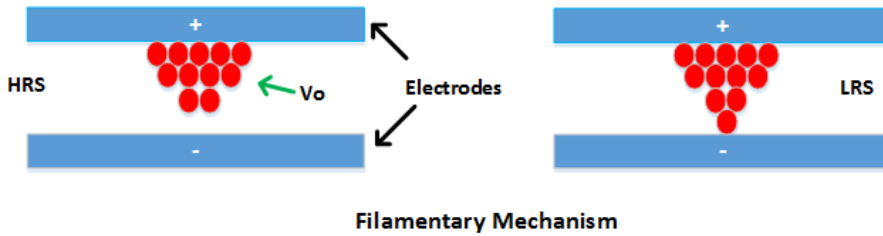


Figure 6.2: Memristor switching mechanism based on conductive filamentary switching, the oxygen vacancies ( $Vo$ ) form a filament between the two electrodes, changing the resistance state of the device

This thesis will mainly consider the conductive filamentary model for the analysis of process variability and reliability concerns, in the memristive memories. This is because the filamentary model has recently attracted many considerations in academia and seems to be the fundamental resistive switching mechanism.

Moreover, there exists research works that explain the process variability and endurance failure mechanism of ReRAM devices according to the characteristics of the CF model. However, due to importance of the ionic drift this Chapter of thesis will also consider it in some sections, such as in the write time analysis of memristive devices.

## **6.3 Reliability Concerns in Memristor Devices (ReRAM)**

Although memristive devices seem to be promising candidates adequate for various areas of application such as in memories and neuromorphic chips [13], they still face challenges to be solved before becoming a mainstream element. In this sense, this thesis analyzes two of the main reliability concerns in the design of memristive memories, including process variability, endurance degradation and briefly studies the Random Telegraph Noise (RTN).

### **6.3.1 Process Variability**

Memristive devices are affected by significant variations in their parametric characteristics, where statistical variation in their high and low resistance values could be a major barrier for reliable operation of this device. These statistical deviations can be categorized into two types: device-to-device and cycle-to-cycle variability, where the first one characterizes the uniformity inside a memory array and the former characterizes the time-varying device stability [117].

This thesis have focused on the device-to-device variability due to its deterministic behaviour. Regarding this, and firstly considering ionic drift model (HP-model as the first memristive model), the ohmic conduction model of the resistance value can be estimated as Eq. 6.7:

$$R = \rho \times \frac{L}{S} \quad (6.7)$$

where due to the nano-scale size of these devices there could exist variations in

## CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES

area ( $S$ ) and memristor thickness ( $L$ ) from the manufacturing process [118]. Note that also fluctuations in the doping concentration of oxygen vacancies can cause an additional source of variability inside the  $\rho$  parameter.

The presence of the process variability in the behavior of ionic drift memristive model has been analyzed by Matlab simulation. For instance, Figure 6.3 depicts a 200 sample Monte-Carlo analysis by using the HP memristor model, where LRS and HRS values contain variation.

The mean values are considered as:  $LRS = 100\Omega$  and  $HRS = 16K\Omega$  (taken from [40]), and we assume  $\sigma = 10\%$  variation in LRS values and  $\sigma = 20\%$  variation in HRS values, as fresh devices. Figure 6.3 also demonstrates how the response hysteresis  $I - V$  loop, the memristor current and state variable ( $x$ ) changes due to the resistance variations.

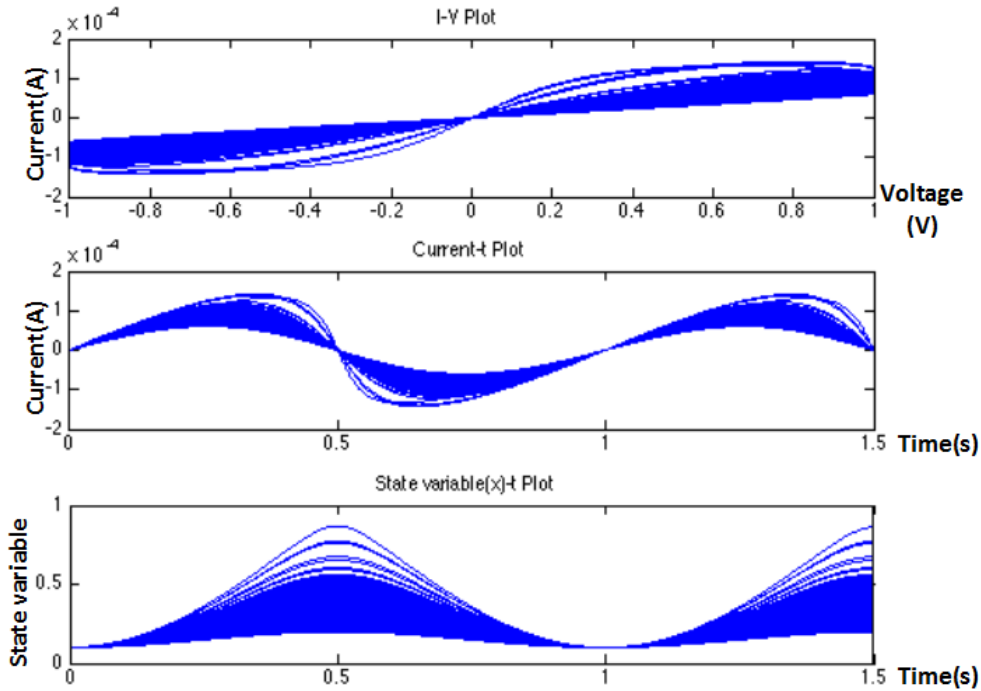


Figure 6.3: Monte-Carlo analysis of HP memristor model while considering variability effect in HRS and LRS values

Now then if the model is changed to the more recent memristive model

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

of conductive filamentary, then it was noted that an initial random electroforming process is needed to form a weak filament in device. This random manufacturing step could be a major source for the device-to-device variability, due to creating different size filaments and various channel size in each device [119].

It is reported in the literature that LRS variations is due to the variation of number and size of CFs [16], while the HRS variation is because of deviations in CFs length. Therefore, due to the dependence of the tunneling current in the tunneling distance, generally HRS values have higher variations than the LRS ones [120], and for this, they are a bigger concern in reliability analysis scenarios.

Furthermore, the variability is also affected by the operation parameters, such as voltage, pulse width and temperature [121]. Increasing the first two would reduce the resistance variation while higher temperature would increase it. This device-to-device variability will cause read instabilities by reducing the read margin and deviations in write time and write energy.

Note that, the graphs in Figure 6.3 could not tell us how reliable a memristor is in storing and reading the values in memories. In this thesis, we will assume a normal distribution for both HRS and LRS values, with a defined mean and standard deviation value.

### 6.3.2 Endurance Failure Mechanism

Another reliability concern in ReRAM devices is the limited numbers of write cycles called endurance. This mechanism depends on different parameters, among others, the environment temperature and switching speed. It is worth noting that temperature and high voltage can accelerate the endurance degradation [16].

The endurance failure mechanism can be better clarified by a detailed analysis of the set and reset process in ReRAM devices. The SET process in ReRAM devices is correlated with a soft breakdown of the resistive switching layer. The oxygen ions ( $O^{2-}$ ) and oxygen vacancies ( $Vo$ ) are generated by the electric field in the setting phase. Then, the  $O^{2-}$  get drifted to the anode and

## CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES

---

the existing  $Vo$  constructs a conducting filament, and the resistance value switches from high to low.

However, the recombination of  $O^{2-}$  and  $Vo$  will rupture the filament and cause a switch from low to high resistance, which is called the RESET process [122]. Due to the degradation mechanisms, the distance between high and low resistance values cannot remain like the fresh device, and their values would get variations from their expected value.

In the literature, three types of endurance failure are reported [24]. The first one (Figure 6.4I) is related with the HRS value shrinks in contrast to the LRS values that increase, due to the oxidation at electrode interfaces (generally anode-electrode) during the process or forming step.

The second failure type (Figure 6.4II), considers the HRS degradation by decreasing its value because of extra oxygen vacancies ( $Vo$ ). These extra  $Vo$  can make the filament stronger, and therefore it would be more difficult for the RESET voltage to rupture the filament.

Finally, the third wear-out mechanism (Figure 6.4III), also impacts the HRS values, according to lack of  $O^{2-}$  to recombine with  $Vo$ . This could be due to consumption of  $O^{2-}$  stored in the electrode layer, where the restored  $O^{2-}$  during SET cannot be equal to consumed  $O^{2-}$  during the RESET phase. Figure 6.4 [24] shows these three endurance failure mechanisms.

It has been reported that by optimizing the set and reset voltages the endurance degradation can be postponed and the number of cycles can get extended in a few orders of magnitude [123]. Nevertheless, the degraded device cannot recover unlimited number of times because the  $O^{2-}$  would be consumed during the device operation, and the conductive filament cannot be successfully ruptured by  $Vo$  and  $O^{2-}$  recombination.

In Chapter 7 of this thesis, endurance degradation type I is chosen for modeling, where also the possible limited recovery in resistive switching devices is not considered.

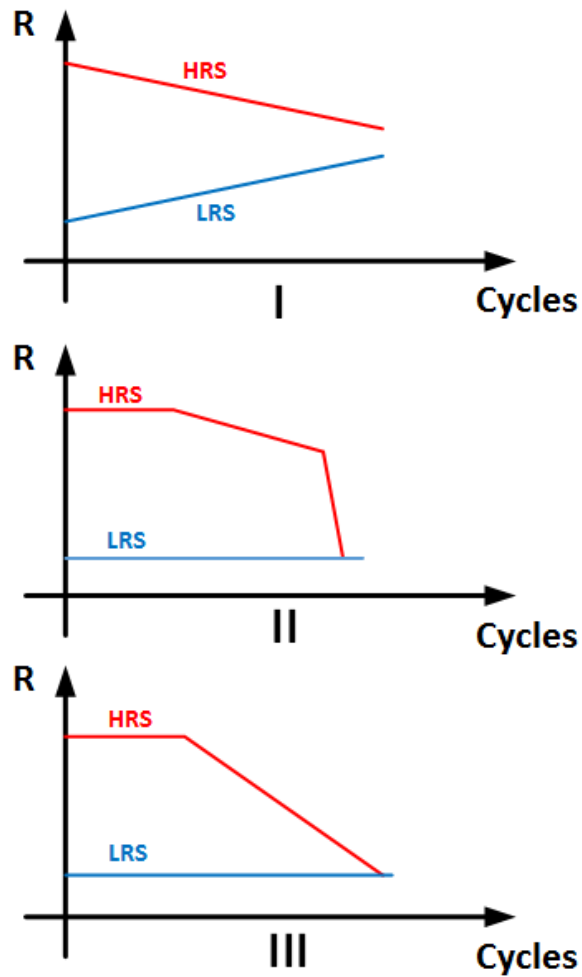


Figure 6.4: Endurance failure behavior in memristive devices [24]

### 6.3.3 Random Telegraph Noise (RTN)

Random Telegraph Noise (RTN) is a noise phenomenon often seen in semiconductor devices, such as MOSFETs, p-n junctions, metal contacts and Metal-Insulator-Metal (MIM) junctions, etc. It causes discrete random fluctuations between constant values, and has become a significant issue in advanced nano-scale circuit design [124].

The RTN effect is due to the capture and emission of charge carriers near the interface and can cause variations of threshold voltage ( $V_T$ ) and drain cur-



## CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES

rent ( $I_d$ ) in MOSFETs. In ReRAM devices the RTN effect generates current fluctuations at high and low resistance values due to activation/deactivation of the electron traps inside the filament [124]. The current variation by RTN can induce read instabilities and reduces the memory read window in ReRAM memories if enough consideration is not taken care.

RTN is usually described by Markovian process [125], where its switching process is assumed at only two discrete values. The distance between these two values is called RTN peak to peak ( $RTN_{p,p}$ ) and is the maximum noise, which might occur. Figure 6.5 depicts one example of RTN noise in ReRAM current in which two states are considered. Its behavior in this figure comes from the fact that in each reset/set cycle of the device where the filament is constructed and ruptured, it can have different structure and therefore fluctuations of current may exist.

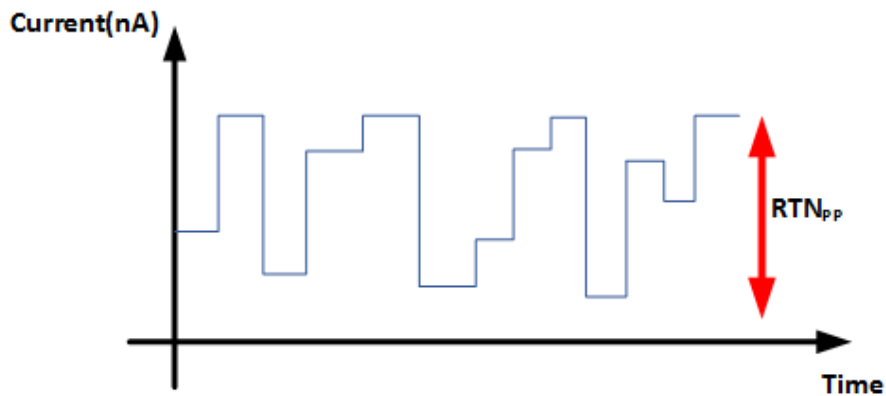


Figure 6.5: Current fluctuations in ReRAM because of RTN

This thesis will mainly analyze the impact of the two reliability concerns of process variability and endurance degradation in the function of the ReRAM memory, and RTN has not been included as objective analysis of this thesis.

## 6.4 Impact of Reliability Concerns in Memristive Memories

In this section, several simulation results obtained by Matlab simulations are presented to evaluate the existing significant reliability concerns in the memristor memories.

The objective is to analyze the potential impact of process variability and endurance degradation in the robust operation of the memory array. This study first evaluates the probability of error ( $Pe$ ) in the read operation while reading the memristor state (at LRS or HRS state). Afterwards, it will analyze the evolution of probability of error also in the write process of a HP memristor.

### 6.4.1 Probability of Error in a Read Operation

There are different mechanisms to read the stored values in memristive cells; however all are based on measuring the memristor state in form of current or voltage and comparing it with a reference value [126][127]. Depending whether the memristor is in LRS or HRS state, the current/voltage would differ and a '0' or '1' would be read from the cell.

In this sense, to consider the process variability in the two resistance states (LRS and HRS) of the memristive devices, two normal distributions for both values are assumed in such a way that the ratio between their mean values is around 2X-5X, (this is common in Multi-Level storage Cells (MLC) and also in aged devices) and plot the probability of error while reading the memristor state.

The  $Pe$  is a variable, which determines the likelihood of an incorrect read in the memristive memory cell in function of a reference resistance value ( $R_{th}$ ), with which the reference resistance is compared. The  $Pe$  graph is plotted while considering a reference point in resistance value (in which below  $R_{th}$  it is expected to be in LRS state and for higher than  $R_{th}$  the HRS state is anticipated) and sweep it along the two distributions corresponding to LRS and HRS.

## CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES

For instance, Figure 6.6a depicts the two LRS and HRS distributions (e.g. with mean values= $1K\Omega$ ,  $2K\Omega$  and  $\sigma = 100\Omega$ ,  $200\Omega$  for LRS and HRS respectively). Then, Figure 6.6b also shows the  $Pe$  evolution versus the threshold point, that differentiates the LRS and HRS modes. The  $Pe$  plot shows the best option to choose as the reference point and obtain the minimum probability of error ( $Pe_{min}$ ), while reading the memristor value. Note that the x-axis in the Figure 6.6a is the resistance value ( $R$ ), which is different from the x-axis in the Figure 6.6b that means the reference resistance point ( $R_{th}$ ).

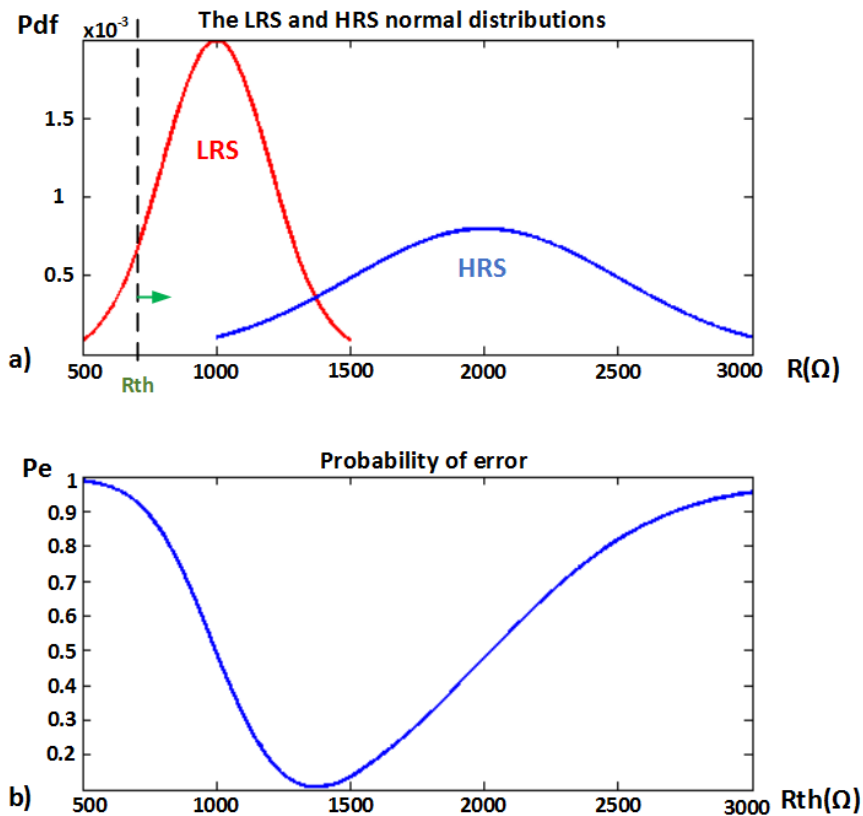


Figure 6.6: a) HRS and LRS normal distributions and the reference point ( $R_{th}$ ) sweeping along them, b)  $Pe$  according to the corresponding value of reference resistance

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

Next, in order to analyze the impact of variability and aging simultaneously, it is considered that the two LRS and HRS distributions move toward each other as a consequence of degradation as it could be observed in Figure 6.7a. Then, the evolution of the  $Pe$  in function of the threshold resistances graph is analyzed in Figure 6.7b, and as expected, the  $Pe_{min}$  value would raise because of the device aging and the fact that both (HRS and LRS distributions) shift toward each other. This would emphasize the importance of considering the simultaneous impacts of variability and aging for robust and reliable memristive memory design.

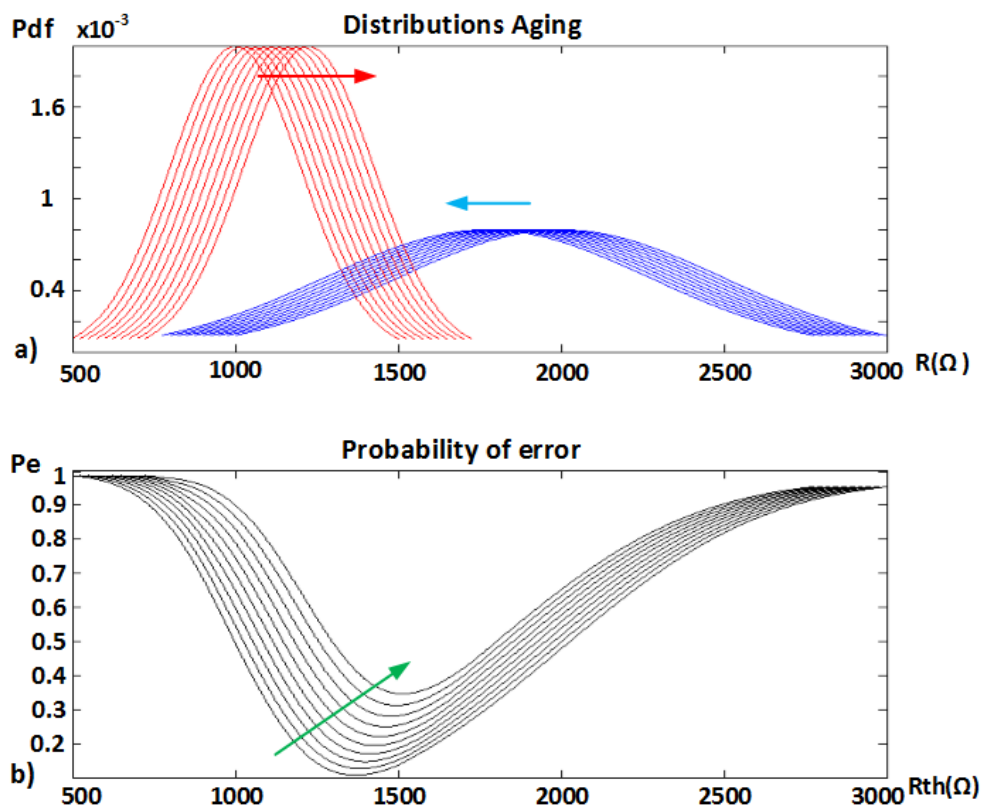


Figure 6.7: a) HRS and LRS distribution move toward each other because of aging, b)  $Pe$  worsens by the aging

Now let's consider another two normal distributions of LRS and HRS resistance random values from other experimental results in memristive memo-

## CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES

---

ries to analyze the reliability of a memristive cell in a read operation. Therefore, we characterized the resistance values with obtained numbers from experimental results in the literature [24][128]:  $\mu(LRS) = 1K\Omega$ ,  $\mu(HRS) = 100K\Omega$  and  $\sigma(LRS) = \sigma(HRS) = 20\%$  of the mean value corresponding to their mean and standard deviation values.

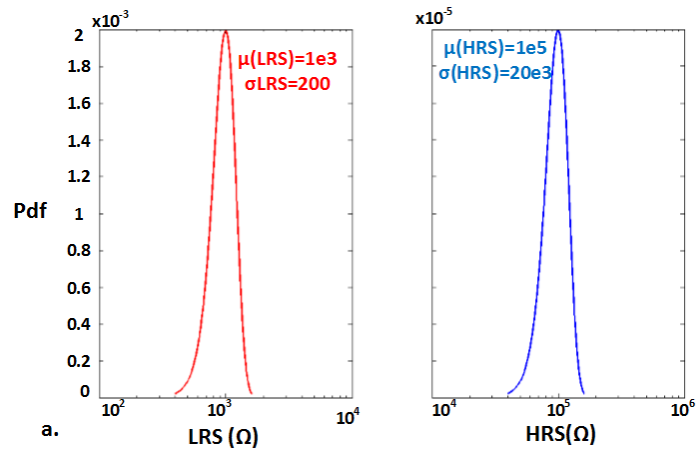
Figure 6.8a shows the two truncated LRS and HRS probability distributions of fresh devices between  $-3\sigma$  to  $+3\sigma$  with these new values. Again the Pe graph is plotted while sweeping the reference resistance value along the two LRS and HRS distributions.

It is observed that as the  $\mu(HRS)$  and  $\mu(LRS)$  values get closer to each other (getting their ratio smaller, caused by degradation), the Pe in the read operation of a memristive cell becomes higher. Therefore, Figure 6.8b points out the relevance of considering the simultaneous impact of variability and endurance degradation for robust and reliable memristive memory design.

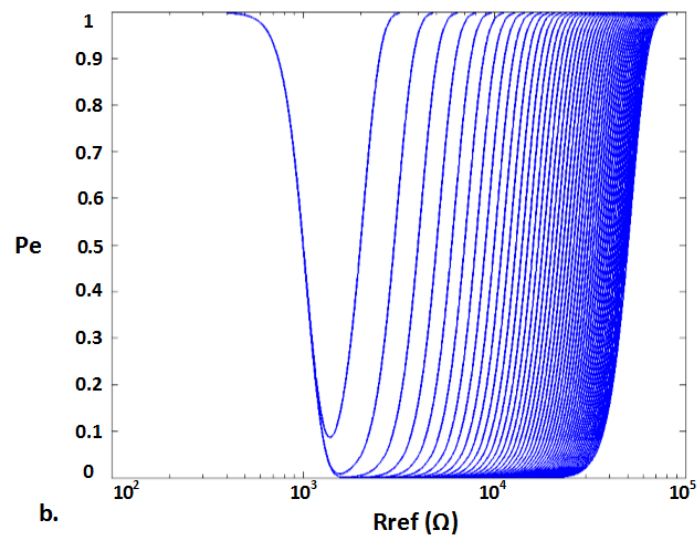
Now then, after analyzing the reliability in read operation, let's consider the impact of variability and endurance degradation in the write operation of a HP model memristor.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---



(a)



(b)

Figure 6.8: a) The LRS (left) and HRS (right) probability distributions following a normal distribution b) Pe in function of different Rref values, the closer the two distributions the Pe getting worse

### 6.4.2 Probability of Error in a Write Operation

There are two techniques for the writing of the memristor devices [50]. The first approach is called preset writing, where the selected memristor is initially reset to the LRS state and then by applying an appropriate pulse it is

## CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES

---

switched to the desired state.

The second method is based on iterative writing and reading pulses until the memristor is written to the proper value. By applying a squared pulse voltage to the memristor, with amplitude  $V_A$  and pulse width  $Tw$ , we can change the state of memristor.

The required duration of pulse ( $Tw$ ) to assure the change of the state of memristor (from LRS to HRS and vice versa) in the ionic drift memristor model has been calculated in [129] and can be expressed as follows in Eq. 6.8:

$$Tw = \left( \frac{D^2}{HRS \times LRS \times 2 \times \mu \times V_A} \right) \times (HRS^2 - LRS^2) \quad (6.8)$$

Due to memristor process variability this write time differs from cell to cell, and therefore would cause error in the case of a limited pulse width. In this work, variability in LRS and HRS values are considered with the values similar to the previous section.

In this context, 10,000 Monte-Carlo simulations are run in Matlab to obtain the distribution of  $Tw$ , and then to calculate the probability of the write error in respect to sweep of a reference  $Tw$  (variable  $Tw_{ref}$ ). In this sense, Figure 6.9a presents the obtained probability density function of  $Tw$ , meanwhile variations in LRS and HRS values are considered following the distributions mentioned in previous sections.

The generated  $Tw$  distribution are fitted with a Gamma distribution bounded to  $[0, \infty]$ . This distribution is typically used to model aging and time-varying degrading statistic mechanisms as it starts from zero and continues to infinity.

Next, in Figure 6.9b, the probability of error in the write process is presented, while sweeping the  $Tw_{ref}$  along the probability density function. It is shown as the  $Tw_{ref}$  increases (the pulse becomes wider) the probability of the error reduces and converges toward zero, evidently with extra energy cost.

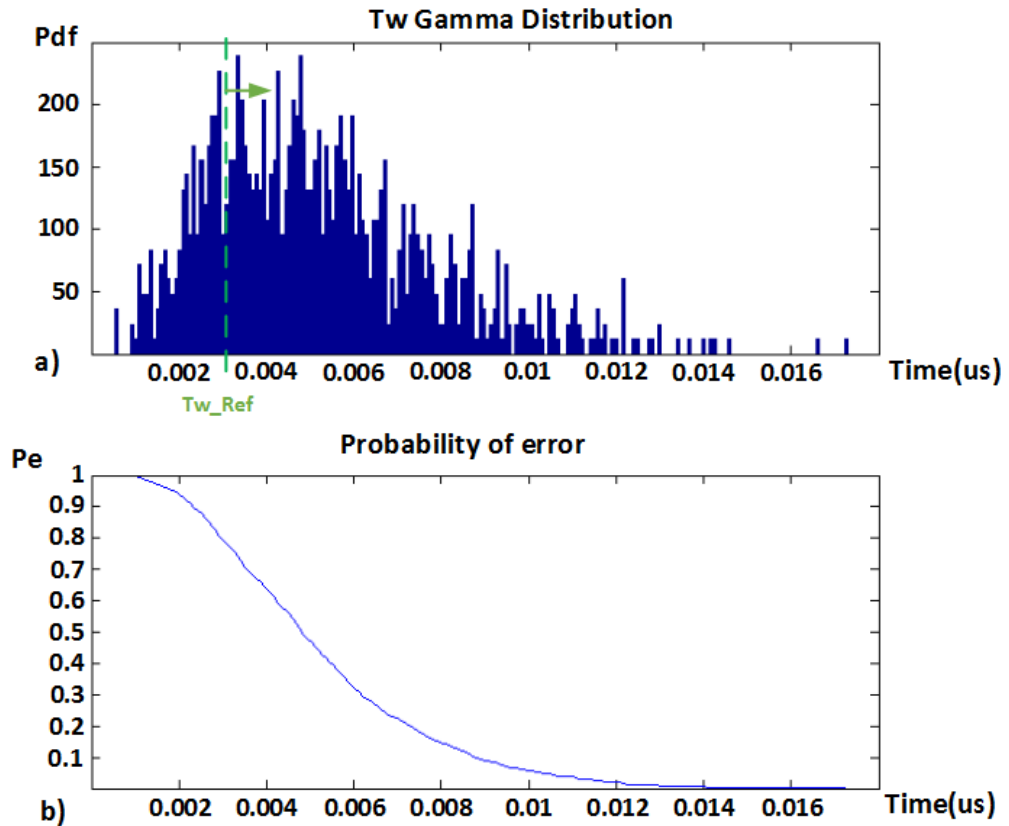


Figure 6.9:  $T_w$  according to variability introduced in LRS and HRS values, b)  $P_e$  graph according to the sweep of the  $T_{wref}$  point

## 6.5 Summary and Conclusions

This Chapter of thesis has first analyzed the memristive devices (ReRAM) switching mechanisms, and chosen the HP-model and CF model as the main ones for further evaluations.

Second, it analyzes the two main potential reliability effects in these devices, including process variability, endurance failure mechanism and briefly introduces RTN behavior in ReRAMs.

Third and as the main contribution of this Chapter, some outcomes from the parameter variation in memristive memories have been presented, which



## **CHAPTER 6. PROCESS VARIABILITY AND RELIABILITY CONCERNS IN MEMRISTIVE MEMORIES**

---

would increase the probability of failure of the memory system in the read and write operation, shown through Matlab simulations.

The simulation results show that the process variability and endurance degradation would increase the probability of the error in the read phase, while in the writing cycle a relation between the writing failure probability and the writing pulse duration, is presented.

All this motivates the necessity for design of adaptive and dynamic circuit design techniques to overcome such vulnerabilities, which is the topic of following chapters in this thesis.

The results of this Chapter has been published in Conference papers of International Workshop on CMOS Variability (VARI 2014) [130] and International Conference on Memristive Systems (MEMRISYS 2015) [131].



# 7

## Memristive Crossbar Lifetime Evaluation and Reconfiguration Strategies

### 7.1 Introduction

Once we have analyzed the memristor reliability, in Chapter 6, here after we would analyze the behavior of memristors-based systems, when they are subjected to the different reliability aspects. Then, the aim of these studies is to estimate the lifetime of a single memristor device analytically in terms of endurance cycles. Moreover, the lifetime of a set of memristive devices constructing a crossbar will be also evaluated analytically, up to the first and second failure appearance.

In the following, two reconfiguration approaches will be introduced to extend the crossbar lifetime. It will be shown through these two approaches how they can extend the system lifetime, where utilizing an adaptive method can be more efficient in using the most of resources uniformly.

## 7.2 Memristive Lifetime Modeling

In Chapter 6 it was shown that the device-to-device process variability in resistance states of memristive devices could be considered as two independent normal distributions. Then, three different endurance failure behaviors were introduced (see Section 6.3.2, LRS and HRS converging toward each other).

This Chapter will consider the first type (Figure 6.4I), as the most significant in memristive device lifetime, because it can reduce the device lifetime significantly. Like in Chapter 3, where the BTI aging was modeled by linear equations, here similarly, the endurance failure mechanism can be modeled with a linear degradation of HRS and LRS toward each other [24] as it is shown in Figure 7.1.

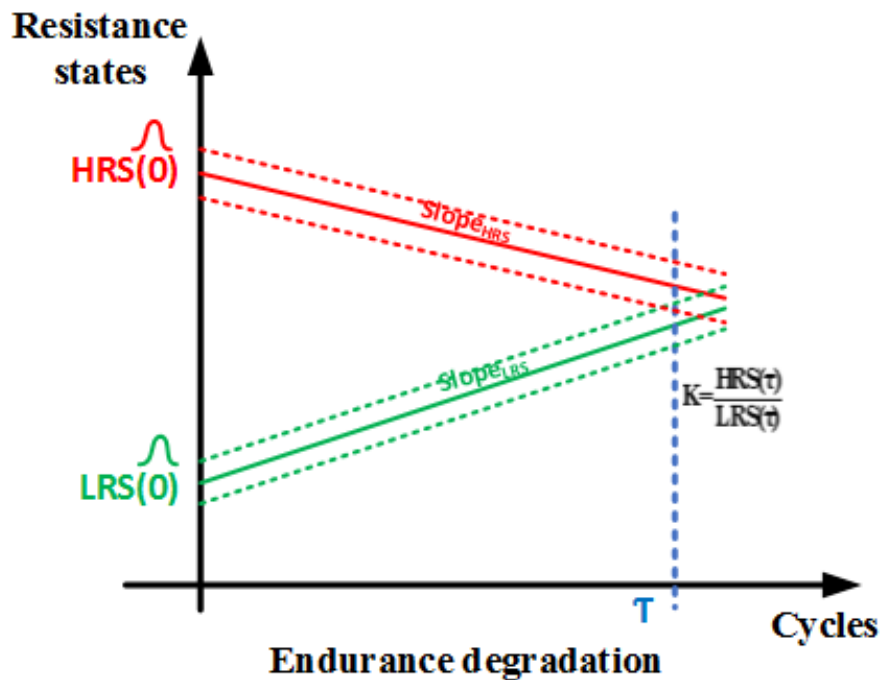


Figure 7.1: Endurance degradation behavior in HRS and LRS values shown together with process variation at origin. The point ( $\tau$ ) is where the ratio of HRS to LRS reaches the critical point in terms of reliability and is equal to  $K$

Note that, there could exist variations at the origin, due to the process

## CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME EVALUATION AND RECONFIGURATION STRATEGIES

---

variability. So then, the memristor lifetime ( $\tau$ , a random variable) is evaluated by assuming a linear approximation for the degradation slopes of LRS and HRS with the number of cycles following the concept shown in Figure 7.1. The values for these slopes, as well as  $HRS(0)$  and  $LRS(0)$  (which are the initial resistance values at cycle zero), are all taken from experimental measurements [24] [128].

Then, by defining the point of critical care for a memristor as the point where the  $\frac{HRS(\tau)}{LRS(\tau)}$  ratio becomes equal to a given  $K$  value (as shown in Figure 7.1), the point of failure is derived as in Eq. 7.1:

$$K = \frac{HRS(0) - SlopeHRS \times \tau}{LRS(0) + SlopeLRS \times \tau} \quad (7.1)$$

Solving Eq. 7.1 would result in obtaining the memristor lifetime expression, measured in number of endurance cycles Eq. 7.2.

$$\tau = \alpha \times HRS(0) - \beta \times LRS(0) \quad (7.2)$$

Where  $\alpha$  and  $\beta$  are coefficients that depend only on the slopes of degradation ( $slopeHRS$  and  $slopeLRS$ ) and the selected  $K$  parameter as in Eq. 7.3 and Eq. 7.4.

$$\alpha = \frac{1}{SlopeHRS + K \times SlopeLRS} \quad (7.3)$$

$$\beta = \frac{K}{SlopeHRS + K \times SlopeLRS} \quad (7.4)$$

Next, from the principle of sum of independent normal distributions, the mean and variance values for the  $\tau$  variable are calculated as in Eq. 7.5 and Eq. 7.6:

$$\mu(\tau) = \alpha \times \mu(HRS(0)) - \beta \times \mu(LRS(0)) \quad (7.5)$$

$$\sigma^2(\tau) = \alpha \times \sigma^2(HRS(0)) + \beta \times \sigma^2(LRS(0)) \quad (7.6)$$

So, under the selected numerical assumptions from experimental results

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

(e.g.  $\mu(LRS0) = 1K\Omega$ ,  $\mu(HRS0) = 100K\Omega$ ,  $\sigma(LRS0) = \sigma(HRS0) = 20\%$  of  $\mu$ , and taking now  $K=5$  [128]), the numerical lifetime of a single memristor (starting from cycle zero) follows a normal distribution, as it is shown in Figure 7.2, as an example. It shows that considering the above conditions in presence of process variability and endurance degradation a single memristor has an average lifetime equal to 1 million cycles with a standard deviation equal to 80000 cycles.

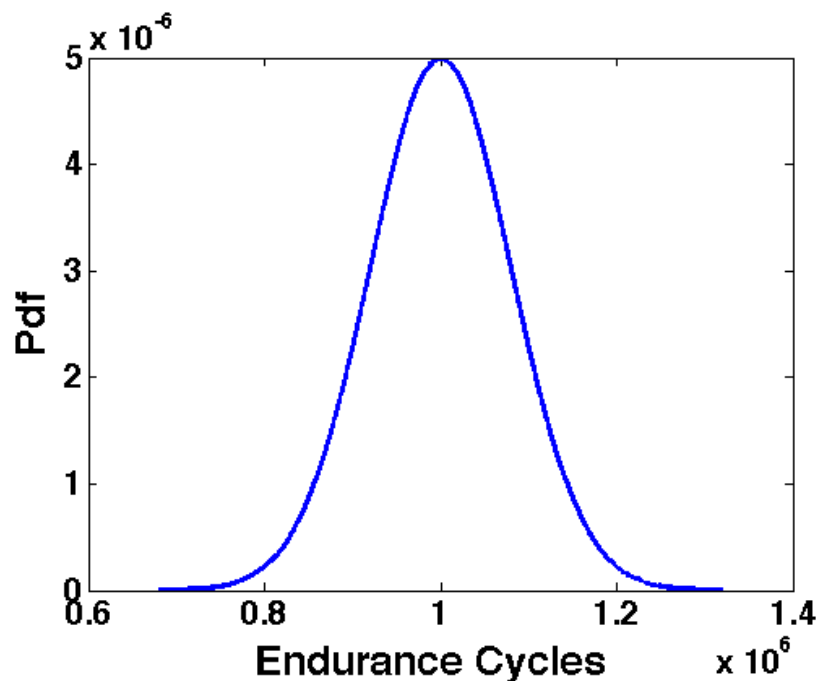


Figure 7.2: The PDF( $\tau$ ) for fresh memristors, the mean value for number of endurance cycles in this normal distribution of  $\tau$  is  $1e6$  and sigma is 80000 in terms of endurance cycles

### 7.3 Crossbar Lifetime Analysis

In the next section the objective is to analytically obtain the probability distribution of the cycles number up to the first and second failure, when a multiple-component crossbar matrix with  $n$  memristors is considered.

## CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME EVALUATION AND RECONFIGURATION STRATEGIES

---

### 7.3.1 Lifetime up to the First Failure

The initial analysis step is focused to find the probability distribution of the number of cycles for the memristor of the crossbar, which first reaches the critical ratio of  $K$ , considering process variability and independent variables for each memristor of the matrix. This can be calculated as the probability distribution of the minimum value of the individual independent random variables ( $\tau_i$ ) in each cell and for the complete  $n$ -component crossbar.

Assuming a set of  $\tau_i$  values with a normal distribution, where  $1 \leq i \leq n$ , the goal is to calculate the probability density function (PDF) of the random variable  $g$ , where  $g = \text{Min}(\tau_1, \tau_2, \dots, \tau_n)$ . This concept means that given  $n$  random variables of  $\tau_i$ , then the probability  $p(\text{min}(\tau_1, \tau_2, \dots, \tau_n) \leq g)$  implies that at least one  $\tau_i$  must be smaller than  $g$ . The probability that at least one  $\tau$  must be smaller than  $g$  is equivalent to one minus the probability that all  $\tau_i$  are greater than  $g$ . Regarding this the Cumulative Distribution Function (CDF) of  $g$  is found as in Eq. 7.7:

$$CDF(g) = 1 - (1 - CDF(\tau))^n \quad (7.7)$$

where  $CDF$  of random variable  $\tau$  (considering normal distribution) is defined as Eq. 7.8:

$$CDF(\tau) = \int_{-\infty}^{\tau} PDF(\tau) = \int_{-\infty}^{\tau} \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} dz \quad (7.8)$$

From Eq. 7.7 and Eq. 7.8 the  $PDF(g)$  can be calculated as in the following Eq. 7.9 and Eq. 7.10:

$$PDF(g) = \frac{\partial CDF(g)}{\partial \tau} \quad (7.9)$$

$$PDF(g) = n \times \left(1 - \int_{-\infty}^{\tau} \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} dz\right)^{n-1} \times \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{(\tau-\mu)^2}{2\sigma^2}} \quad (7.10)$$

Considering same numerical assumptions from the previous section, Figure 7.3 shows the probability distribution of the crossbar lifetime up to the

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

first failure ( $PDF(g)$ ) together with the  $PDF(\tau)$  to be compared. Here,  $n$  (number of memristors in crossbar) is considered to be  $n = 16$  as a matter of example.

Moreover, Figure 7.3 points out the crossbar lifetime is smaller than a single memristor's lifetime, and shows a quasi-gaussian distribution. Note that, the mean and standard deviation of the  $g$  are calculated by numerical calculations verifying the analytical result.

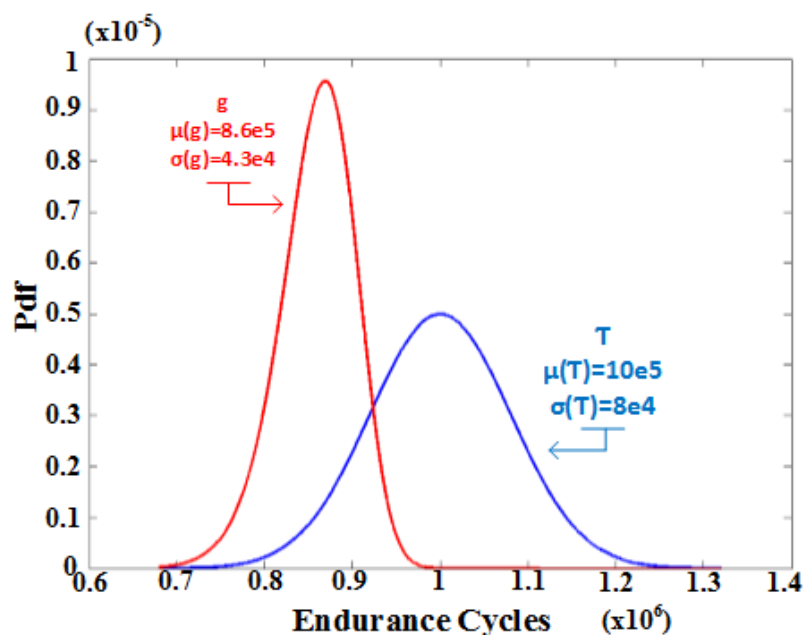


Figure 7.3: The  $\tau$  (individual fresh memristor lifetime) and  $g$  (crossbar lifetime up to first fail) probability distributions

### 7.3.2 Lifetime up to the First Failure Verified with Monte-Carlo Analysis

The correctness of the  $PDF(g)$  obtained and shown in the previous section has been verified by performing 10,000 Monte-Carlo simulations. In each experiment we generate  $n$  random numbers ( $\tau_1, \tau_2, \dots, \tau_n$ ), samples of a normal distribution with a known  $\mu(\tau)$  and  $\sigma(\tau)$ , each one representing the lifetime of a single memristor and then the minimum value among them is found.



## CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME EVALUATION AND RECONFIGURATION STRATEGIES

As shown in Figure 7.4 the  $PDF(g)$  from the Monte-Carlo analysis perfectly matches with our analytic approach. It demonstrates that the crossbar has an average lifetime equal to 860,000 cycles with a standard deviation equal to 43,000 cycles.

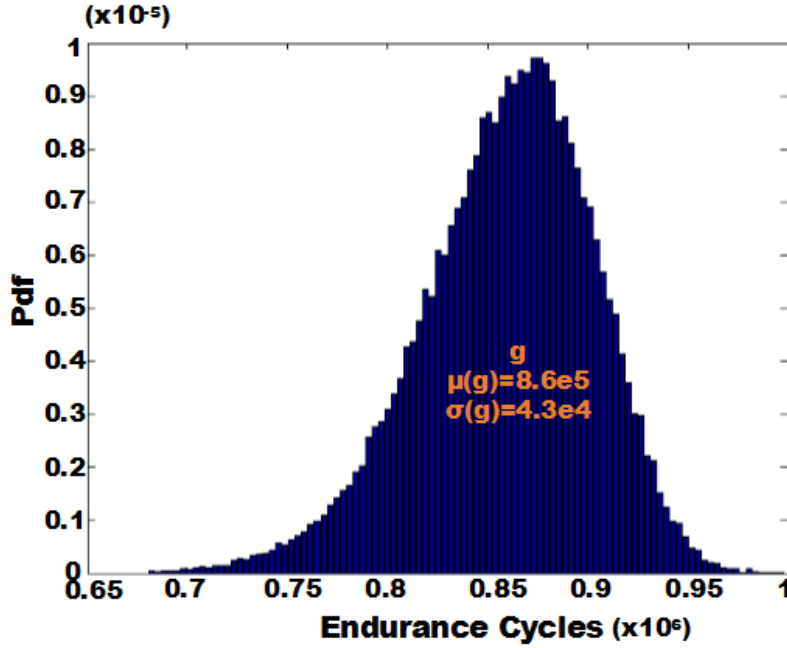


Figure 7.4: The  $g$  probability distribution verified with Monte-Carlo simulation

### 7.3.3 Lifetime up to the Second Failure

In the following the probability distribution of cycles up to the second failure would be calculated. By having the number of cycles at the beginning ( $\tau$ ) and at the point of first failure ( $g$ ), another random variable ( $h$ ) is defined, which is related with the lifetime at cycle zero minus the time of the first failure as in Eq. 7.11:

$$h = \tau - g \quad (7.11)$$

Then, next equation 7.12 presents the mean value for the  $h$  and Eq. 7.13 presents the standard deviation. Note that since  $\tau$  and  $g$  are not independent,

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

random variables the  $\sigma(h)$  is calculated by considering the correlation factor ( $\rho$ ) as in Eq. 7.14:

$$\mu(h) = \mu(\tau) - \mu(g) \quad (7.12)$$

$$\sigma(h) = \sqrt{\sigma^2(\tau) + \sigma^2(g) + 2 \times \rho \times \sigma^2(\tau) \times \sigma^2(g)} \quad (7.13)$$

$$\rho = \frac{\sum_i (\tau_i - \mu\tau) \times (g_i - \mu g)}{\sqrt{\sum_i (\tau_i - \mu\tau)^2} \times \sqrt{\sum_i (g_i - \mu g)^2}} \quad (7.14)$$

Next, obtaining the crossbar lifetime up to the second failure follows the same mathematical principle explained in previous section and is similar to the first failure study. Regarding this the  $PDF(y)$  is found as in Eq. 7.15, when  $y = Min(h_1, h_2, \dots, h_{n-1})$ . It is now  $n - 1$  memristive devices because one memristor is not considered after the first fail.

$$PDF(y) = (n - 1) \times \left(1 - \int_{-\infty}^h \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} dz\right)^{n-2} \times \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{(h-\mu)^2}{2\sigma^2}} \quad (7.15)$$

In this sense, Figure 7.5 shows the probability distribution of lifetime for a crossbar up to the second failure (i.e. number of cycles for the interval between first and second fail). This lifetime distribution is also verified by using Monte-Carlo simulation in Figure 7.6.

The results of crossbar lifetime analysis shown in Figure 7.3 and Figure 7.5 demonstrate that the memristive crossbar lifetime is highly reduced for posterior fails after the first and mainly second failure. This implies the need to establish efficient reconfiguration mechanisms to achieve reliable memristive crossbar applications. In this sense, next section introduces two reconfiguration approaches to extend the crossbar lifetime.

## CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME EVALUATION AND RECONFIGURATION STRATEGIES

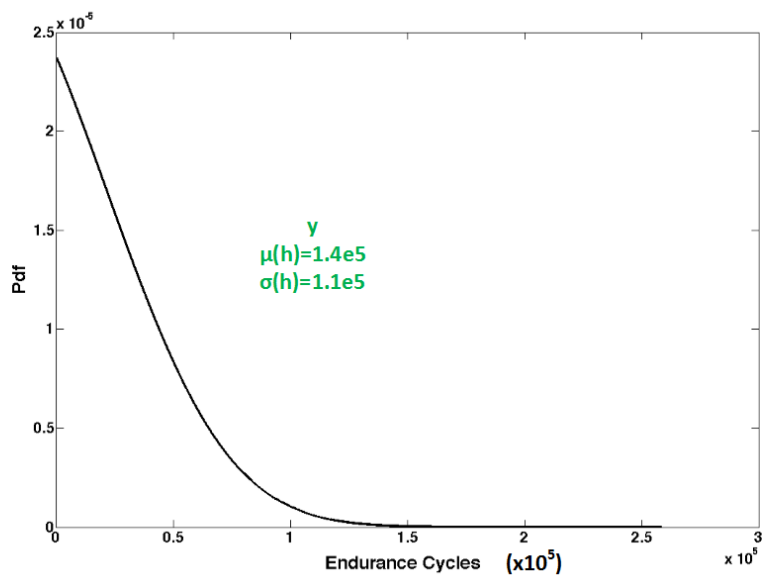


Figure 7.5: The  $y$  (crossbar lifetime between first to second fail) probability distributions

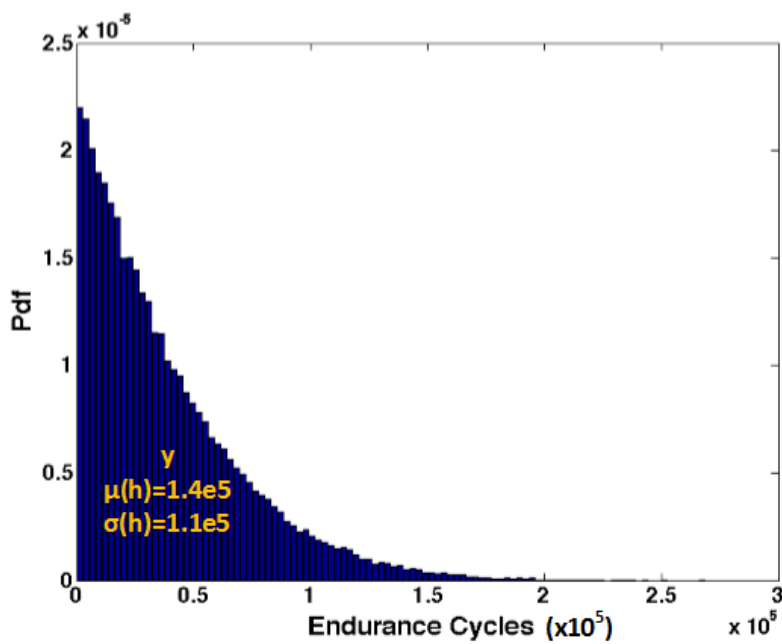


Figure 7.6: The  $y$  probability distribution verified with Monte-Carlo simulation

## 7.4 Non-Adaptive and Adaptive Reconfiguration in Memristive Crossbar

Similarly to SRAM memories (Chapter 4), the reconfiguration techniques could be used to extend the system lifetimes in crossbar memories [132]. The state of the art techniques are based on conventional repair techniques, such as row/column replacement of faulty one with a spare one. In fact, these are approaches, which the use of spare units is limited only to the time that a fail occurs.

Another novel approach can be based on the utilization of spare and operational units together with high simultaneity, through advanced reconfiguration techniques. These present a relevant enhancement of the system performance, in relation of the results observed in previous chapters, and due to this improvement, we have chosen this approach as a baseline configuration. Therefore, in this Chapter two types of reconfiguring techniques named as non-adaptive and adaptive reconfiguring approach are proposed and analyzed.

### 7.4.1 Non-Adaptive Reconfiguration

Let's assume a memristive crossbar of size  $N \times N$ , where only  $m \times m$  units are in active mode to perform a given function. Both  $N \times N$  and  $m \times m$  crossbars are square sized and structured symetrically (squared shape). So, in this non-adaptive reconfiguring approach it is considered that the reconfiguration mechanism skips the whole original  $m \times m$  crossbar, even with most healthy memristors once a memristor in the original  $m \times m$  reaches its lifetime limit. So then, every time a near-failing device is detected the operational  $m \times m$  skips to the next  $m \times m$  crossbar, as it is depicted in Figure 7.7.

So then, in this technique the memristive crossbar lifetime would be solely extended if as many  $m \times m$  unique crossbar structures can be allocated, inside the  $N \times N$  crossbar. This means that when an  $m \times m$  crossbar reaches the end of its lifetime, the whole  $m \times m$  structure shifts inside the  $N \times N$  crossbar. We consider an operative matrix ends its life when a memristor of the matrix

**CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME  
EVALUATION AND RECONFIGURATION STRATEGIES**

---

reaches the end of its working lifetime. Consequently, the lifetime extension is in proportion of number of  $m \times m$  crossbars blocks that can fit inside the  $N \times N$ . If  $N$  is equal to  $\alpha$  times of  $m$  ( $N = \alpha \times m$ ), then  $\alpha^2$  times unique  $m \times m$  crossbars can be allocated inside a  $N \times N$  crossbar and the lifetime extension would be equal to  $\alpha^2$  times of a single  $m \times m$  crossbar lifetime.

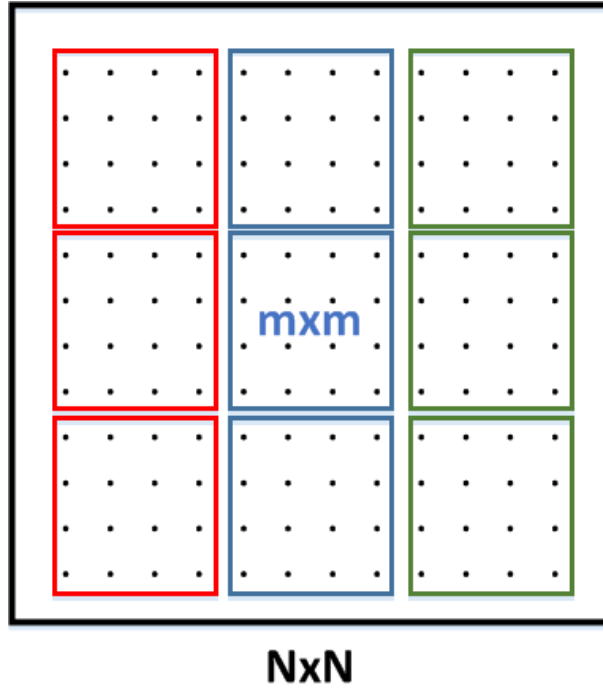


Figure 7.7: Non-adaptive reconfiguring approach of a  $m \times m = 4 \times 4$  in  $N \times N = 16 \times 16$  memristive crossbar, where each shift is a complete  $m \times m$  jump

The non-adaptive technique can extend the crossbar lifetime by getting benefit from high redundancy inside a crossbar. However, the  $m \times m$  lifetime in each sector arrangement is limited to the weakest unit in that structure, and this limits the efficient utilization of the resources both in the local  $m \times m$  and global  $N \times N$  crossbar.

Therefore, in order to optimize the lifetime extension to its higher level, it is needed to use another reconfiguring approach (adaptive reconfiguring), which can perform the shifts more intelligently. This technique is based on a dynamic redundancy allocation strategy that this thesis proposes in the next section.

### 7.4.2 Adaptive Reconfiguration

This technique provides the possibility to use the crossbar resources more uniformly in a more balanced way, in order to extend its lifetime efficiently. In this approach, the mxm structure can shift inside the NxN crossbar, but this time, the shift would be only a given number of columns corresponding to the place of the weakest unit. So then, Figure 7.8 shows that in this strategy the skip is not fixed, but sensitive to the location of the near-failing device.

Note that in the previous approach the shifting step was the whole mxm frame in comparison with now that the new mxm structure can have some nodes in common with the previous mxm structure. For example, if we assume that NxN=16x16 and mxm=4x4 is allocated in the left corner of NxN crossbar, if there is a weak unit (i.e. unit with the lowest endurance, or with the  $\frac{HRS}{LRS}$  value closer to  $K$ ) in column 3 of mxm crossbar which is reaching its lifetime limit, then the mxm crossbar will only shift 3 columns to the left and still utilize one column of the previous mxm structure, what involves a benefit in a more optimum use of the system resources.

There could be different strategies for dynamic shift of mxm inside NxN crossbar, for instance the mxm shift can be in the x-axis or y-axis or even in diagonal direction. In addition, the first mapping of the mxm inside NxN can be in different locations such as in the center or in the left/right corner and this would influence the shifting strategy.

This thesis assumes that the first region for mxm mapping is in the left corner of NxN crossbar, and the mxm shifts in respect to place of the weakest element in the x-axis direction (canonical strategy). When the mxm matrix reaches to the right corner of NxN where it cannot be shifted by maintaining its original size anymore, the mxm crossbar would jump to the first left corner of the NxN crossbar (a complete shift of mxm structure in y-axis direction). In the next phases the shifting would be similar until most of resources in NxN crossbar are utilized. Figure 7.8 describes this procedure for an example (m=4 and N=16).

CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME  
EVALUATION AND RECONFIGURATION STRATEGIES

---

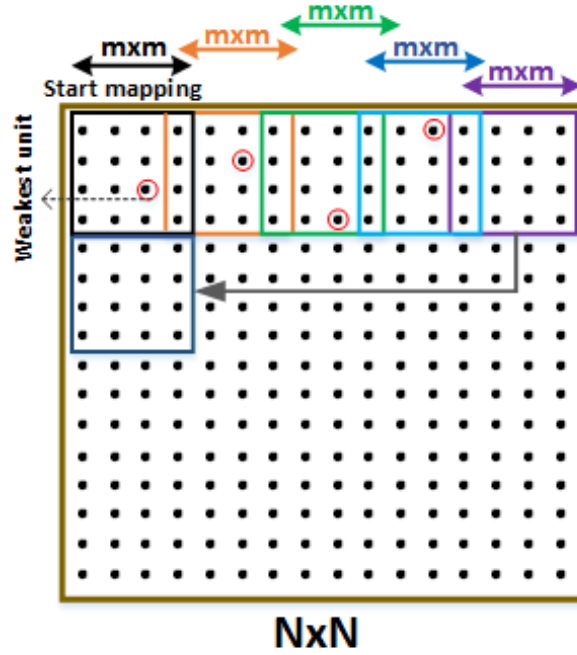


Figure 7.8: Adaptive reconfiguring approach of an  $mxm=4 \times 4$  in  $N \times N=16 \times 16$  memristive crossbar, where each shift is in respect to location of weakest unit

In order to estimate the crossbar lifetime in the adaptive reconfiguration approach, first we determine the expected number of shifts that an  $mxm$  crossbar can make inside an  $N \times N$  crossbar. Each shifting step can vary from 1 to  $m$  depending to the place of weakest element inside  $mxm$ . By solving this problem and knowing the total crossbar lifetime with adaptive reconfiguration an approximation of average lifetime at each  $mxm$  shift inside the  $N \times N$  crossbar can be obtained. If  $T$ =Total crossbar lifetime,  $t_1$ =  $mxm$  lifetime at first configuration,  $t_j$ =lifetime of  $mxm$  at each shift, and  $E$ =expected number of  $mxm$  shifts inside  $N \times N$ , then the total lifetime of  $mxm$  in  $N \times N$  crossbar can be written as Eq. 7.16:

$$T = t_1 + \sum_{j=1}^E t_j \quad (7.16)$$

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

### 7.4.2.1 Determination of the Number of Shifts

The expected number of mxm shifts in NxN can be solved by using different approaches, while considering two different assumptions. The first assumption is based on the fact that each shift has an equal probability, as it can be a random number between 1 to m from uniform distribution and its probability is equal to  $\frac{1}{m}$ . Considering this, the problem is solved mathematically and also by Monte-Carlo simulations, in following sections. **Analytic Approach in uniform-distribution assumption:**

The expected number of shifts can be analytically solved by applying the mathematics in [133] to our case. Here, as an example, assume that m=6, N=12, and therefore, the corresponding number of shifts in each step can be equal to one of values belonging to 1, 2, 3, 4, 5, 6. Note that, here each value of the shift has the same probability of occurrence ( $p = \frac{1}{m}$ ).

Regarding these, the objective is to find the expected number of total shifts that m can make inside N. We can write the following expectations in each step, where for instance  $E(12)=0$  means that the expected number of shifts is 0 if it is already 12.

$$E(12) = 0$$

$$E(11) = 1$$

$$E(10) = 1 + \left(\frac{1}{m}\right)E(11)$$

$$E(9) = 1 + \left(\frac{1}{m}\right)E(11) + \left(\frac{1}{m}\right)E(10)$$

$$E(8) = 1 + \left(\frac{1}{m}\right)E(11) + \left(\frac{1}{m}\right)E(10) + \left(\frac{1}{m}\right)E(9)$$

$$E(7) = 1 + \left(\frac{1}{m}\right)E(11) + \left(\frac{1}{m}\right)E(10) + \left(\frac{1}{m}\right)E(9) + \left(\frac{1}{m}\right)E(8)$$

$$E(6) = 1 + \left(\frac{1}{m}\right)E(11) + \left(\frac{1}{m}\right)E(10) + \left(\frac{1}{m}\right)E(9) + \left(\frac{1}{m}\right)E(8) + \left(\frac{1}{m}\right)E(7)$$

$$E(5) = 1 + \left(\frac{1}{m}\right)E(11) + \left(\frac{1}{m}\right)E(10) + \left(\frac{1}{m}\right)E(9) + \left(\frac{1}{m}\right)E(8) + \left(\frac{1}{m}\right)E(7) + \left(\frac{1}{m}\right)E(6)$$

$$E(4) = 1 + \left(\frac{1}{m}\right)E(10) + \left(\frac{1}{m}\right)E(9) + \left(\frac{1}{m}\right)E(8) + \left(\frac{1}{m}\right)E(7) + \left(\frac{1}{m}\right)E(6) + \left(\frac{1}{m}\right)E(5)$$

$$E(3) = 1 + \left(\frac{1}{m}\right)E(9) + \left(\frac{1}{m}\right)E(8) + \left(\frac{1}{m}\right)E(7) + \left(\frac{1}{m}\right)E(6) + \left(\frac{1}{m}\right)E(5) + \left(\frac{1}{m}\right)E(4)$$

$$E(2) = 1 + \left(\frac{1}{m}\right)E(8) + \left(\frac{1}{m}\right)E(7) + \left(\frac{1}{m}\right)E(6) + \left(\frac{1}{m}\right)E(5) + \left(\frac{1}{m}\right)E(4) + \left(\frac{1}{m}\right)E(3)$$

$$E(1) = 1 + \left(\frac{1}{m}\right)E(7) + \left(\frac{1}{m}\right)E(6) + \left(\frac{1}{m}\right)E(5) + \left(\frac{1}{m}\right)E(4) + \left(\frac{1}{m}\right)E(3) + \left(\frac{1}{m}\right)E(2)$$

$$E(0) = 1 + \left(\frac{1}{m}\right)E(6) + \left(\frac{1}{m}\right)E(5) + \left(\frac{1}{m}\right)E(4) + \left(\frac{1}{m}\right)E(3) + \left(\frac{1}{m}\right)E(2) + \left(\frac{1}{m}\right)E(1)$$



## CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME EVALUATION AND RECONFIGURATION STRATEGIES

---

In this sense, the number of shifts in generalized form can be derived when  $i = 0$  in the Eq. 7.17:

$$E(i) = 1 + \left(\frac{1}{m}\right)[E(i+1) + E(i+2) + \dots + E(i+N)] \quad (7.17)$$

For example, Figure 7.9 shows if  $m \times m = 4 \times 4$  and  $N \times N = 40 \times 40$ , then the average number of shifts for  $m \times m$  in first row block of  $N \times N$  would be equal to 17. This means that from first until the last valid shift it would take 17 steps that the  $m \times m$  structure can shift inside one row block of  $N \times N$ , where  $i$  ranges from  $N-1$  to 0 and  $E(0)$  is equal to average number of shifts. Figure 7.9 depicts the average number of shifts for different values of  $m$  and  $N$ .

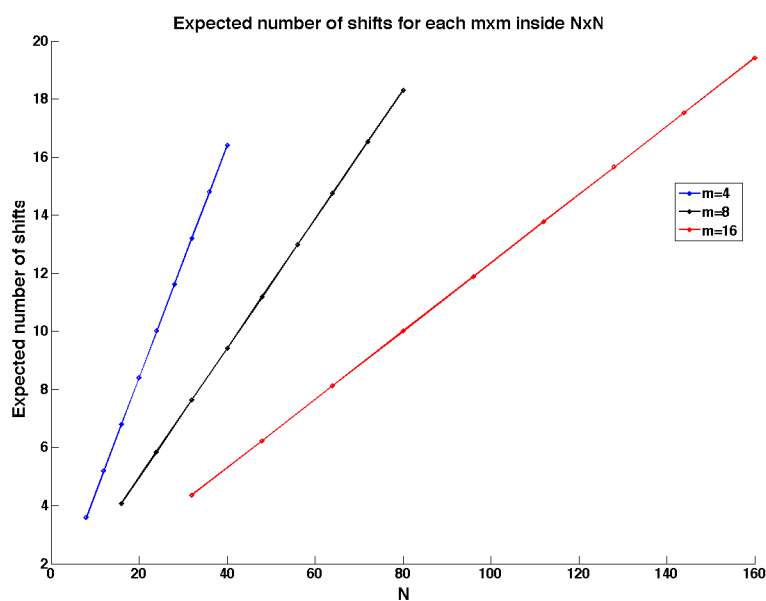


Figure 7.9: Average number of shifts in analytic approach, for  $m \times m = 4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$  and  $N \times N$  ranging from  $8 \times 8$  to  $160 \times 160$

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

### Monte-Carlo Approach in uniform assumption:

In order to verify the mathematical results, we have used Monte-Carlo simulations, as this method can also give us the deviations from calculated numbers. To do so, repeatedly random numbers (representing the possible shifts) are generated between 1 and  $m$  from the uniform distribution, and summed up them together. Once the sum of generated random numbers equals or overflows the value  $N$ , the number of random generations is the expected value for possible shifts of  $m \times m$  inside a row block of  $N \times N$ . This procedure is repeated 10,000 times, and at the end the average of required shifts for a corresponding  $m$  and  $N$  is calculated. In this sense, Figure 7.10 presents the corresponding result for the Monte-Carlo simulation, when the average number of shifts and the corresponding standard deviation is depicted; and the previous mathematical result are verified.

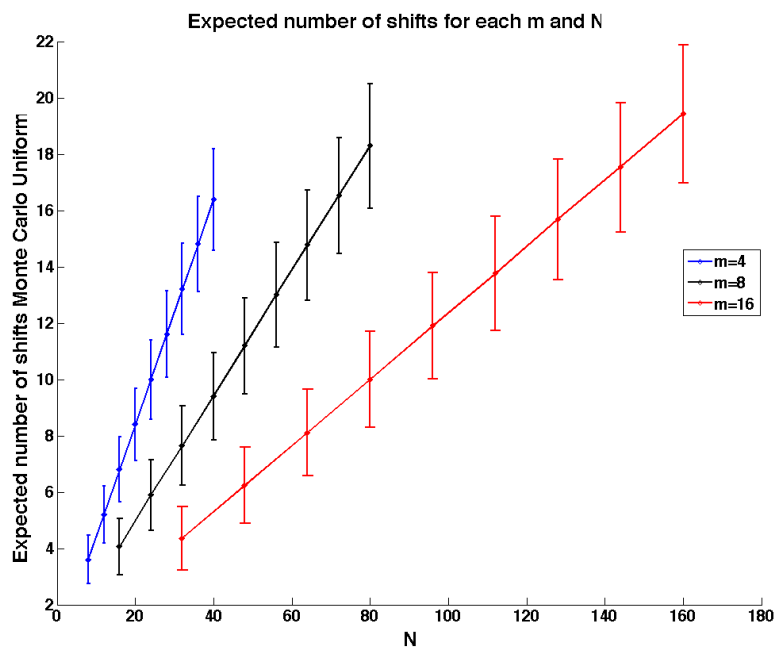


Figure 7.10: Average number of shifts in Monte-Carlo, for  $m \times m = 4 \times 4, 8 \times 8, 16 \times 16$  and  $N \times N$  ranging from  $8 \times 8$  to  $160 \times 160$

## CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME EVALUATION AND RECONFIGURATION STRATEGIES

---

Note that the deviation bars in the Figure 7.10 show the possible deviation from nominal value. Therefore, it can be concluded that the number of shifts do not have big deviations (<15%).

### **Mathematical Semi-Analytic Approach in non-uniform assumption:**

Now, the second assumption is based on the fact that each shift does not have an equal probability. For instance, because of variability or endurance degradation (each shift can be a random number between 1 to  $m$  and its probability now, is not equal to  $\frac{1}{m}$  and can be a random number between 0 and 1). In this context, the problem can be solved by a semi-analytic approach. By performing mathematical analysis now in the semi-analytic approach the Eq. 7.17 can be written as Eq. 7.18:

$$E(i) = 1 + r_1 \times E(i + 1) + r_2 \times E(i + 2) + \dots + r_{N-1} \times E(i + N) \quad (7.18)$$

where  $i$  ranges from  $N-1$  to 0 and  $E(0)$  gives the average number of required shifts. The  $r_1$  to  $r_{N-1}$  are random numbers between 0 and 1. These random numbers are created as following:

1. Create  $m$  random numbers that sum up 1
2. Repeat step one  $\alpha$  times, where  $\alpha = \frac{N}{m}$  and make a set of random numbers ( $r_1$  to  $r_N$ )
3. Use  $N-1$  terms of the above set ( $r_1$  to  $r_{N-1}$ ) for the calculation in (7.18)

Next,  $E(0)$  is evaluated mathematically from 7.18 and the above process (steps 1-3) is repeated 10,000 times, each time with a different set of random portions ( $r_1, r_2, \dots, r_{N-1}$ ). At the end, all  $E(0)$  values are averaged from each iteration and the expected number of shifts in this scenario is obtained. Figure 7.11 depicts the expected number of shifts in the semi-analytic approach and compares it with the analytic approach.

It is observed that the analytic approach would give the upper bound for the average number of shifts, which means the optimistic number for

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

the possible shifts a mxm structure can have inside a NxN crossbar. As an example, if mxm=4x4 and NxN=40x40, then the average number of shifts for mxm in one row block of NxN would be equal to 13 in the semi-analytic approach while in the analytic approach the number of shifts is equal to 17.

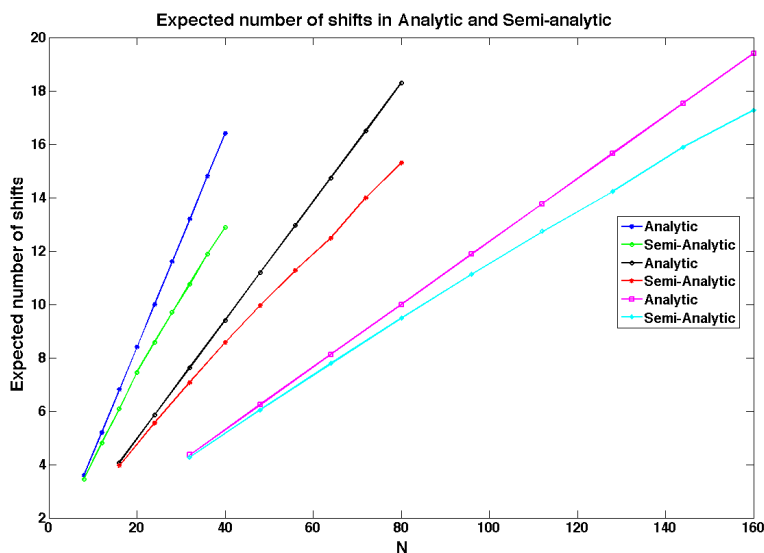


Figure 7.11: Average number of shifts in semi-analytic, for mxm=4x4, 8x8, 16x16 and NxN ranging from 8x8 to 160x160

Therefore, the semi-analytic is an approximation of the number of shifts that an mxm structure can make inside an NxN crossbar in presence of possible variations (i.e. process variation, endurance degradation). Thus, to find approximations of the average mxm lifetime in the crossbar for each shift, we have first computed the total mxm lifetime by Monte-Carlo simulation, and then divided its value to the number of shifts. Therefore, Table 7.1 presents the results for such calculation in different realizations of mxm structure. It is observed that, as the mxm structure gets bigger the lifetime average per shift gets smaller, due to larger number of shifts for bigger mxm structures.

## CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME EVALUATION AND RECONFIGURATION STRATEGIES

---

Table 7.1: Average lifetime per shift in mxm structures

	mxm=4x4	mxm=8x8	mxm=16x16
Average lifetime per shift (in cycles)	$6.23 \times 10^5$	$5 \times 10^5$	$4.2 \times 10^5$

### 7.4.3 Comparison Between Non-adaptive and Adaptive Approaches

In order to find the crossbar lifetime extension of an mxm matrix inside an NxN crossbar, by using the adaptive technique, a Monte-Carlo approach (with 10,000 iterations), is utilized and the results are analyzed with Matlab. So then, the analysis steps are as following:

1. Generate a memristive crossbar size=NxN with fresh endurance values and random process parameters (obtained before,  $\mu(\tau) = 10e5$  and  $\sigma(\tau) = 1e5, 2e5, 3e5, 4e5$ ).
2. Allocate the first mxm structure in the left corner of the crossbar.
3. Find the location and the value (number of cycles) of minimum endurance value (weakest memristor) in the mxm structure.
4. Shift the mxm structure toward right in accordance with the location of the weakest element (i.e. minimum endurance), and update the endurance values inside the crossbar regarding the previous aging step.
5. Pursue shifting the mxm structure in the NxN crossbar until the translocation procedure is valid. After each shift update the crossbar endurance values according to endurance degradation in previous steps.
6. Total crossbar lifetime (maximum endurance for mxm inside NxN) in the adaptive approach is equal to sum of endurance values in each shift.

To obtain the crossbar lifetime in the non-adaptive approach, the crossbar in step 1 is used and possible number of mxm structures is allocated inside

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

it. Then, the crossbar lifetime in the non-adaptive approach would be equal to the sum of minimum endurance values in each mxm structure.

In the next analysis step, Figure 7.12a and Figure 7.12b compare the lifetime extension between both the adaptive and non-adaptive reconfiguring approaches for mxm structures inside NxN (the mean values in percentage and the standard variations in the deviation bars). So then, these figures show the results for various standard deviation (std) values of memristor lifetime (in terms of endurance cycles) in diverse crossbar structures (different values of mxm and NxN).

In this sense, both figures (Figure 7.12a and 7.12b) demonstrate that the adaptive reconfiguring approach can extend the crossbar lifetime more than the non-adaptive one. It also shows that as the number of N increases the benefit of our adaptive approach enlarges and also higher values of standard deviation lead to larger lifetime extension. Furthermore, as the mxm matrix gets bigger (more freedom as a consequence of dynamic shift, in contrast to the non-adaptive one) the lifetime extension has increased.

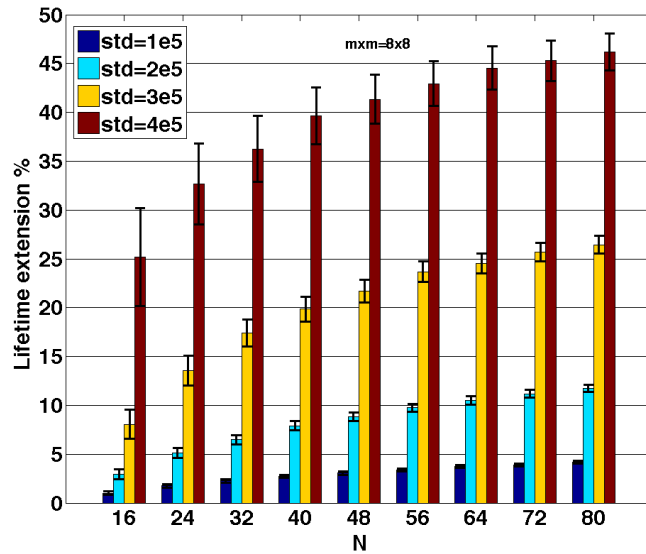
Note that here again the deviation bars show the deviations from calculated average number and the figures demonstrates that the deviations in lifetime improvement are not big. The differences between the non-adaptive versus the adaptive reconfiguring approach are also shown through a 3D bar graph example regarding the better resource usage of the adaptive proposal.

In this sense, Figure 7.13a depicts one row block of NxN crossbar with fresh endurance (non-used) values as bars. Note that, there is some variation in fresh endurance values, because of process variability. Next, Figure 7.13b presents the crossbar state at the end of its lifetime after the consequent endurance degradation, when non-adaptive reconfiguration is used.

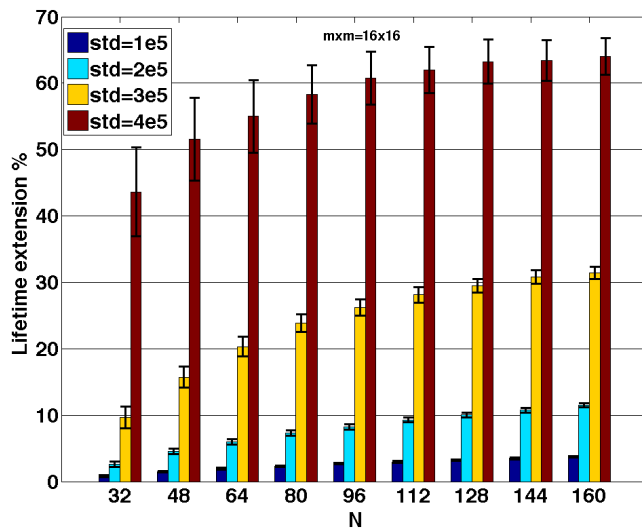
Finally, Figure 7.13c shows the results after the use of adaptive strategy. It is observed that in contrast to the non-adaptive approach, the last proposal depicts that resources are used more efficiently, as their endurance value is more close to the end (fewer memristors with remaining endurance), i.e. system resources are better managed. This is due to the fact that in this technique the shifting step adapts itself to its weakest unit. In this example, mxm=4x4 and NxN=20x20 ( $\mu(\tau) = 10e5$  and  $\sigma(\tau) = 2e5$ ) is considered.

**CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME  
EVALUATION AND RECONFIGURATION STRATEGIES**

---



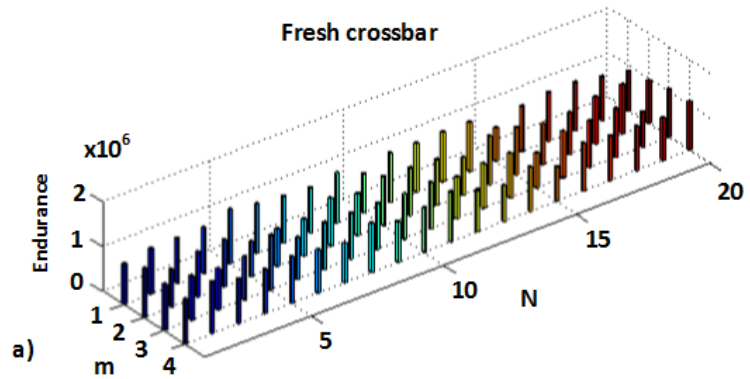
(a)



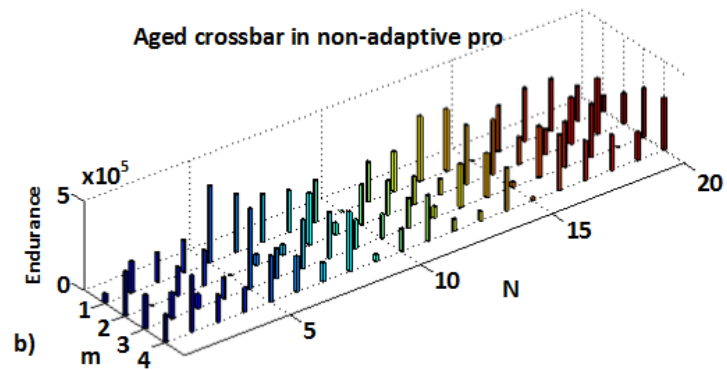
(b)

Figure 7.12: Lifetime extension in adaptive reconfiguring approach versus non-adaptive with various standard deviation (std) values and different array configurations: a)  $mxm=8 \times 8$  and  $N \times N$  ranges from  $16 \times 16$  to  $80 \times 80$  is considered, b)  $mxm=16 \times 16$  and  $N \times N$  ranges from  $32 \times 32$  to  $160 \times 160$  is considered

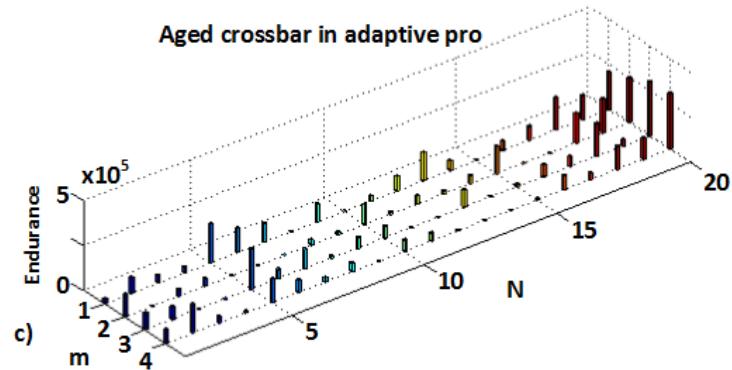
## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS



(a)



(b)



(c)

Figure 7.13: 3D display of one row-block in  $N \times N = 20 \times 20$  crossbar, when a) Fresh crossbar is affected by process variability, b) Lifetime of an aged crossbar when non-adaptive reconfiguration is used, c) Lifetime of an aged crossbar when adaptive reconfiguration is applied



## CHAPTER 7. MEMRISTIVE CROSSBAR LIFETIME EVALUATION AND RECONFIGURATION STRATEGIES

---

These results show the benefits (more balanced aging and larger lifetime increase) of our adaptive approach in front of the non-adaptive one. They also verify the lifetime extension results previously observed in Figures 7.12a and 7.12b , and signify the utilization of an adaptive approach in memristive crossbars by considering device variability and endurance degradation.

Moreover to better evaluate the efficiency of adaptive approach in front of the non-adaptive one, a parameter is defined as Resource Usage Factor (RUF) as in Eq. 7.19:

$$RUF = 1 - \frac{\sum(\textit{Lifetime in remaining devices})}{\textit{total lifetime at time 0}} \quad (7.19)$$

Then, a Monte-carlo simulation with 10,000 iterations has been performed with similar parameters to the previous example, ( $m \times m = 4 \times 4$  and  $N \times N = 20 \times 20$  ( $\mu(\tau) = 10e5$  and  $\sigma(\tau) = 2e5$ )) to obtain the RUF factor in each case. The results show the range of  $RUF = 30 - 40\%$  for non-adaptive reconfiguring approach and  $RUF = 50 - 60\%$  for the adaptive one, expressing again the benefits of adaptive shifting technique in memristive crossbar memories.

### 7.5 Summary and Conclusions

This Chapter of thesis has analyzed the lifetime of memristive devices and crossbars. Therefore, its corresponding contributions include: first proposing an approach to model the endurance degradation and process variation and to estimate the lifetime of a memristive device in terms of endurance cycles.

Second, a statistical approach is presented to predict the lifetime of crossbar up to the first and second failure. It was shown that the crossbar lifetime can be affected significantly by process variation and endurance degradation and therefore reconfiguring approaches are needed. Therefore, the third is the proposal for two advanced reconfiguration approaches to be applied in memristive crossbar memories named as non-adaptive and adaptive reconfiguring methodologies.

Finally, these two reconfiguring approaches were compared and it was shown that the adaptive approach can extend the crossbar significantly and

## **RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS**

---

utilize the resources more efficiently and for instance up to 60% lifetime extension in comparison with non-adaptive approach was shown in different realizations of mxm and NxN.

The results of this Chapter has been published in Conference papers of International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS 2015) [134] and IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH 2015) [135].

# 8

## Monitoring Approach in Memristive Memories with Reconfiguration Mechanisms

### 8.1 Introduction

In the previous chapters (6 and 7) of this thesis we have considered memristive crossbar memories from an abstract point of view. In this Chapter, the objective is to analyze them from a circuit-level point of view and propose specific architectures.

Regarding this, first, a memristive crossbar memory with functional write and read processes is exposed. Afterwards, several monitoring schemes are evaluated to be utilized in the corresponding architecture with non-adaptive and adaptive reconfiguration strategies inside the crossbar. Finally, circuit simulations are presented to verify the memristive crossbar operation and monitoring procedures.

## 8.2 Memristive Crossbar Architectures

Remember that memristive memory systems are commonly organized in a matrix-like structure called crossbar. The storage cell in the crossbar can be built with only one memristor device (1R cell), but due to the well-known problem of sneak-paths (related with the leakage paths in the unselected devices of the crossbar), what will degrade the output read signal in a specific selected cell and might induce error [136]. For this, usually, the bit storage cell is constructed by utilizing complementary devices.

One globally accepted proposed alternative is to use a selecting device such as a transistor, being the name of the bit-cell (1T1R) [50]. The 1T1R cell generally consists of an NMOS transistor and a resistive switching device (e.g. based on a resistive material such as  $HfO_2$ ), then, in this structure the memristor current is correctly controlled through the crossbar.

In this sense, each memristor is turned 'on' or 'off' based on the row-address in the crossbar. Moreover, 1T1R cell has been chosen for the circuit analysis and implementation since it is CMOS compatible, to manufacture the cell in existing fabrication process. Additionally, this makes the crossbar completely sneak path-free, and improves the noise margin, and moreover it is widely used in other recent research works [126][127][137].

Regarding this, Figure 8.1 presents a memristive crossbar memory constructed with 1T1R as storage cells. Each 1T1R cell can be written and read by applying the appropriate signals through the bitline (BL), wordline (WL) and the select line (SL). Next, the write and read processes are exposed for a 1T1R cell inside the crossbar. Note that, a bipolar memristive device is considered in all circuit schematic and simulations.

## CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

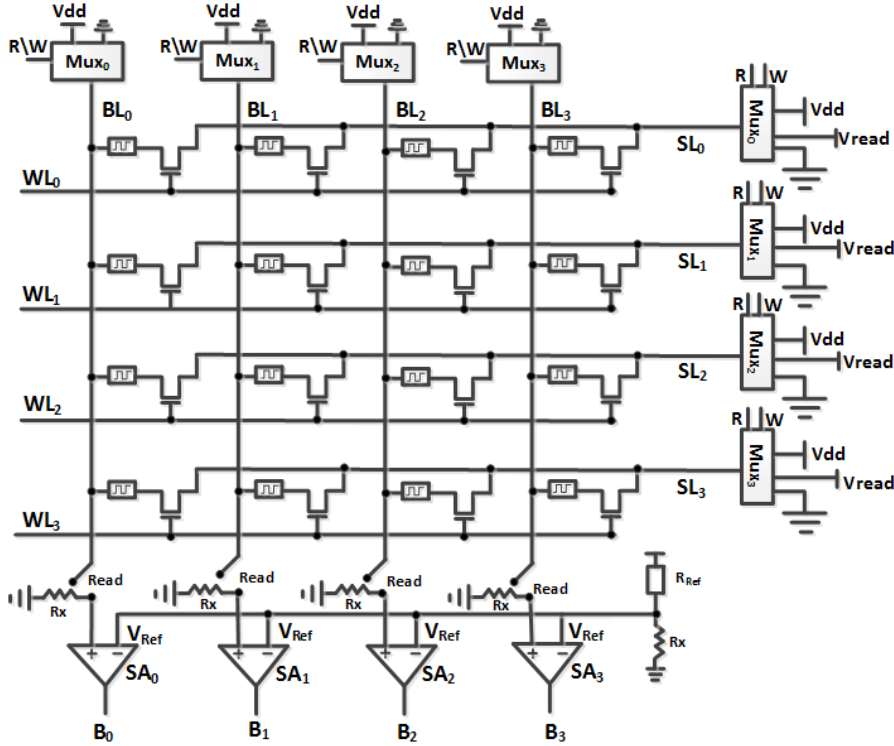


Figure 8.1: A circuit implementation with memristive memories in crossbar configuration

### 8.2.1 Write/Read Cycles

Next, like all memory systems, the write and read cycles should be defined. Therefore, in this section we show the procedure used to perform write and read operations, when 1T1R cell is regarded.

#### 8.2.1.1 Write '1' Cycle (SET)

Writing '1' in the cell, also called SET process, consists on the state change of the respective memristor from HRS to LRS for the selected 1T1R cell. To perform a SET operation first the corresponding WL is activated in the crossbar, and next while the SL is grounded an appropriate voltage is applied at the BL ( $V_{DD}$ ). In this sense, Figure 8.2 shows the corresponding voltages configuration to write '1' in a single 1T1R cell and the current path through the memristor device ( $I_{MEM}$ ).

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

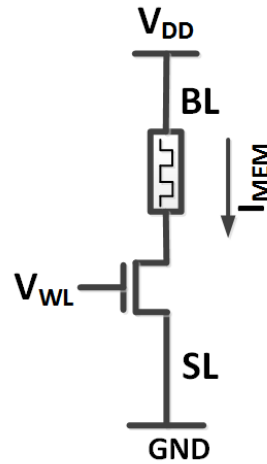


Figure 8.2: Writing 1 inside the 1T1R cell

### 8.2.1.2 Write '0' Cycle (RESET)

Changing the state of memristor from LRS to HRS, is called the RESET process, and it is equivalent to writing a '0' in the cell. For this process the WL is again activated, when a proper voltage ( $V_{DD}$ ) is applied at the SL, and while the corresponding BL is grounded, the other BLs in the block are all in high impedance mode. Figure 8.3 shows the corresponding voltages scheme to write '0' in a single 1T1R cell.

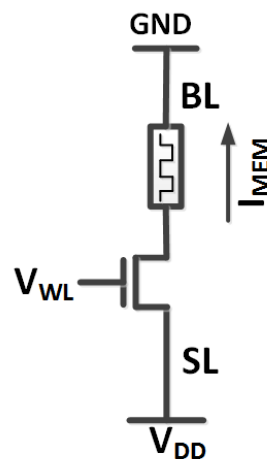


Figure 8.3: Writing 0 inside the 1T1R cell

## CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

---

### 8.2.1.3 Read Cycle

The read process in a memristive crossbar can be implemented by using two different methods; while one is based on voltage, the other one is based on current. The former needs an extra resistor before the sense amplifier, as it is shown in Figure 8.1.

In this approach, a  $V_{read}$  voltage (lower than the write voltage) is applied to the corresponding 1T1R cell, and the BL voltage (a voltage division of the read voltage through the memristor resistance and the resistor  $R_x$ ), will be detected through the comparator giving the state of memristor and bit-cell. A reference cell, consisting of a reference resistance ( $R_{ref}$ ), is used to generate a reference voltage to be compared in the comparator. A resistive device in LRS state will produce a higher voltage in comparison with one at HRS mode. Note that the read voltage ( $V_{read}$ ) is much lower than the write voltage  $V_{DD}$  in order not to modify the state of memristor in the read operation.

Moreover, regarding the read mode based on the current (Figure 8.4), a small current ( $I_{read}$ ) is injected through the BL to the corresponding 1T1R cell and the memristor voltage is sensed and compared with a reference voltage, produced by a reference cell, through a sense amplifier. For instance, Figure 8.4 shows a 2x2 crossbar utilizing the read mode based on the current.

The square boxes in this figure are multiplexers, which govern the appropriate voltage or current to be applied to the cells. Now then, in this thesis the read mode based on voltage is chosen, the reason is controlling the applied voltage to the memristive cell is easier and also the model which we will later use for circuit simulation works better in voltage mode.

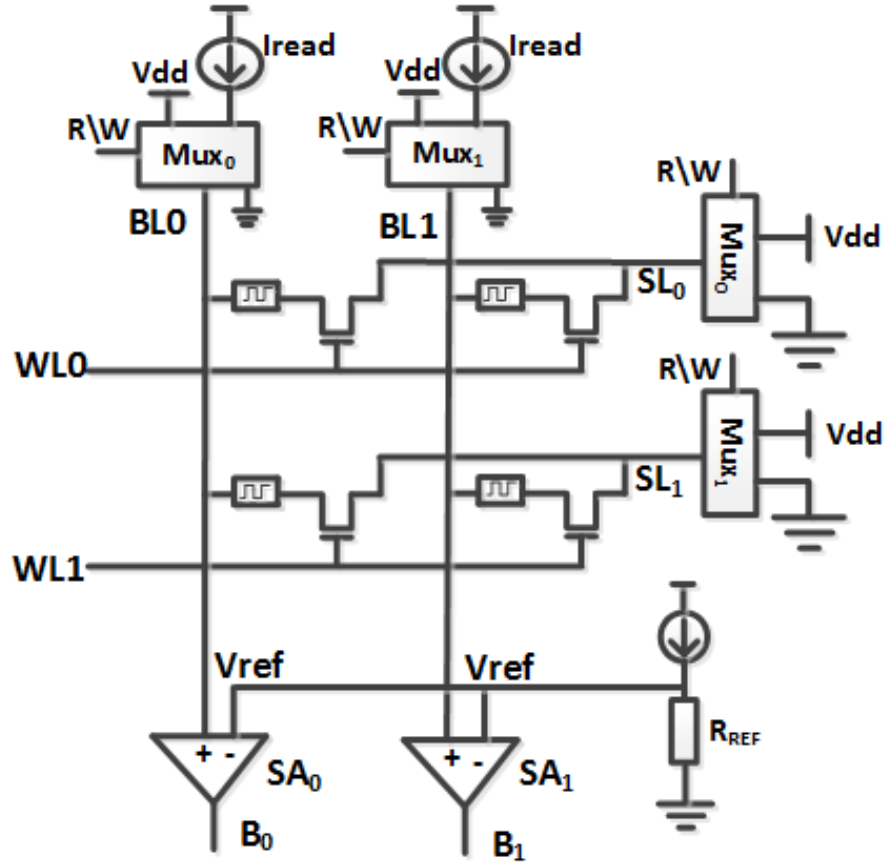


Figure 8.4: Read mode based on current

### 8.3 Monitoring of Crossbar Memristors

In order to detect the faulty cell in a crossbar memory, few recent works have been published in the area of the memristive cells monitoring. In this sense, [138] proposes a modified version of March test to identify the faulty memristors. In [139] a testing technique for open defects in memories based on resistive devices is presented, through analyzing their write access time and write voltage. Finally, [140] introduces a monitoring approach based on sneak path. Sneak paths are undesired paths for current parallel to the intended path in the purely memristive crossbar and [140] uses this property to monitor the memristive cells and detect the faulty ones. All these monitoring schemes have complex circuits and determine fault detection modes.



## CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

---

In this section a novel and simple monitoring approach is introduced to monitor endurance degradation and process variation in memristive cells as well as faulty behavior. This approach can identify efficiently the weakest cell in the operational subset crossbar and guide the reconfiguration flow.

### 8.3.1 Monitoring Scenarios in Crossbar

There could be different scenarios to implement the monitoring procedure in crossbar memories. One approach is based on the objective of making the monitoring procedure independent of the read/write phases inside the crossbar not interfering with them. This would impose added hardware to the crossbar architecture in order to isolate the cells from the normal operation during the monitoring.

The second approach could be based on the fact that there exists some idle cycles inside the memory, which would allow the monitoring procedure to be applied. Therefore, no special isolation would be required and the area overhead is less. This monitoring procedure includes writing 1 and 0 in the cell and checking every time the HRS and LRS values.

In the following, first, the monitoring procedure is analyzed considering the operation isolation scenarios and in different granularities of cell-by-cell, row-by-row, column-by-column, and afterwards the second approach based on existing idle cycles would be explained.

#### 8.3.1.1 Cell-by-Cell Monitoring

We assume as objective to monitor a specific selected cell as shown in Figure 8.5, while the other cells inside the crossbar should function normally (a read/write operation in non-selected cells, meanwhile monitoring the specific cell simultaneously). Observe the memory cells are isolated from the bitlines by added monitoring bitlines and through a multiplexer. Additionally, a separate unit (the box on the upper left side of bitlines) provides the necessary signals for bitline monitor, in this way, the bitline that the monitoring is applied through it, does not interfere with the operational (the one for read/write) bitline.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

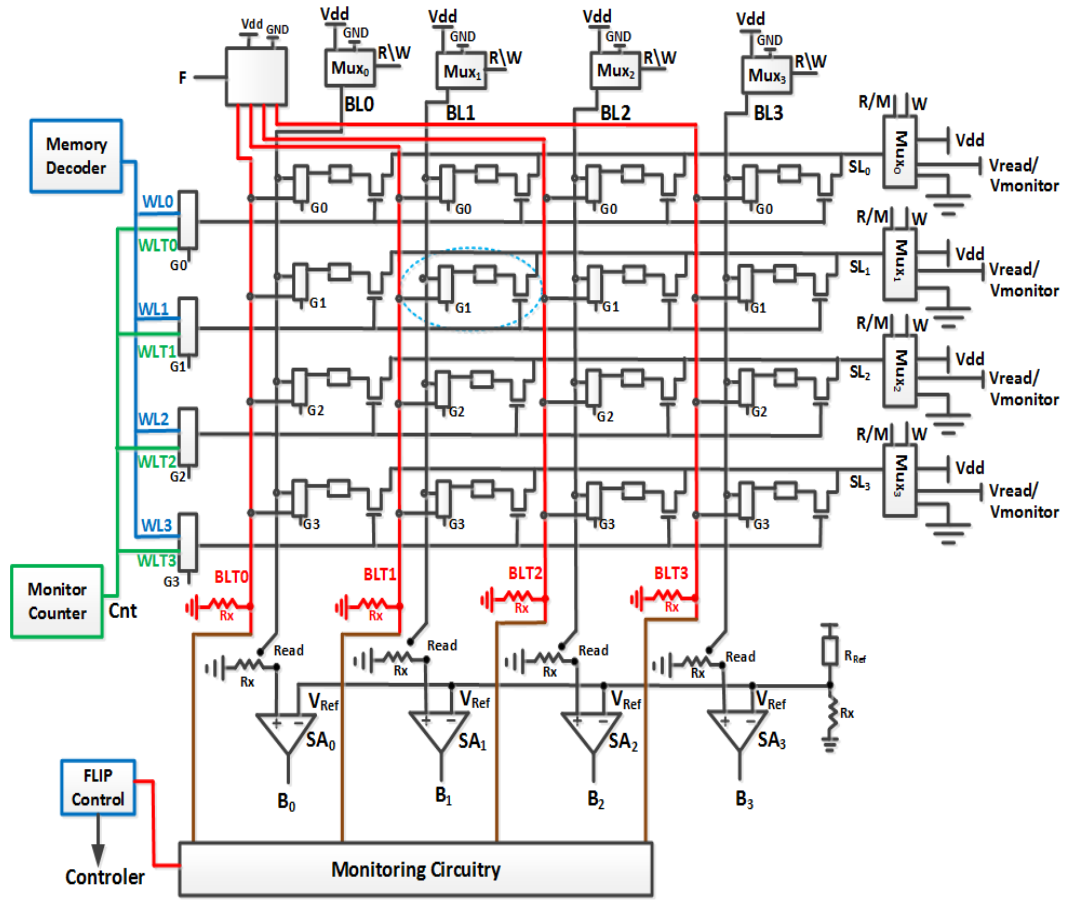


Figure 8.5: Monitoring in crossbar cell-by-cell approach

The monitoring procedure has two phases, measurement and flip. In the former, the selected memristor's resistance at HRS or LRS is evaluated. Afterwards, in the flip phase, the cell state is flipped and the resistance of the other state of the memristor is monitored. Therefore, Table 8.1 shows the applied voltages and signals for both monitor and flip phases the following voltages and signals are applied.

Regarding the mentioned steps for the monitoring, it is clear that although the cells are isolated from the bitline, but still in order to monitor a specific cell the SL gets busy. Therefore, another set of multiplexers is needed to isolate the cells from the SLs, as well.

## CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

Table 8.1: Applied signals in the monitor and flip phase of the cell under monitoring

Phase	BLT1	WLT	SL
Monitor	GND	Activated	V <sub>monitor</sub>
Flip	V <sub>DD</sub> /GND	Activated	V <sub>DD</sub> /GND

### 8.3.1.2 Column-by-Column Monitoring

Now, we assume to monitor the cells column-by-column inside the crossbar, as shown in Figure 8.6 (dotted line). Then, it is needed to monitor all the cells inside the chosen column, while the other cells can function normally.

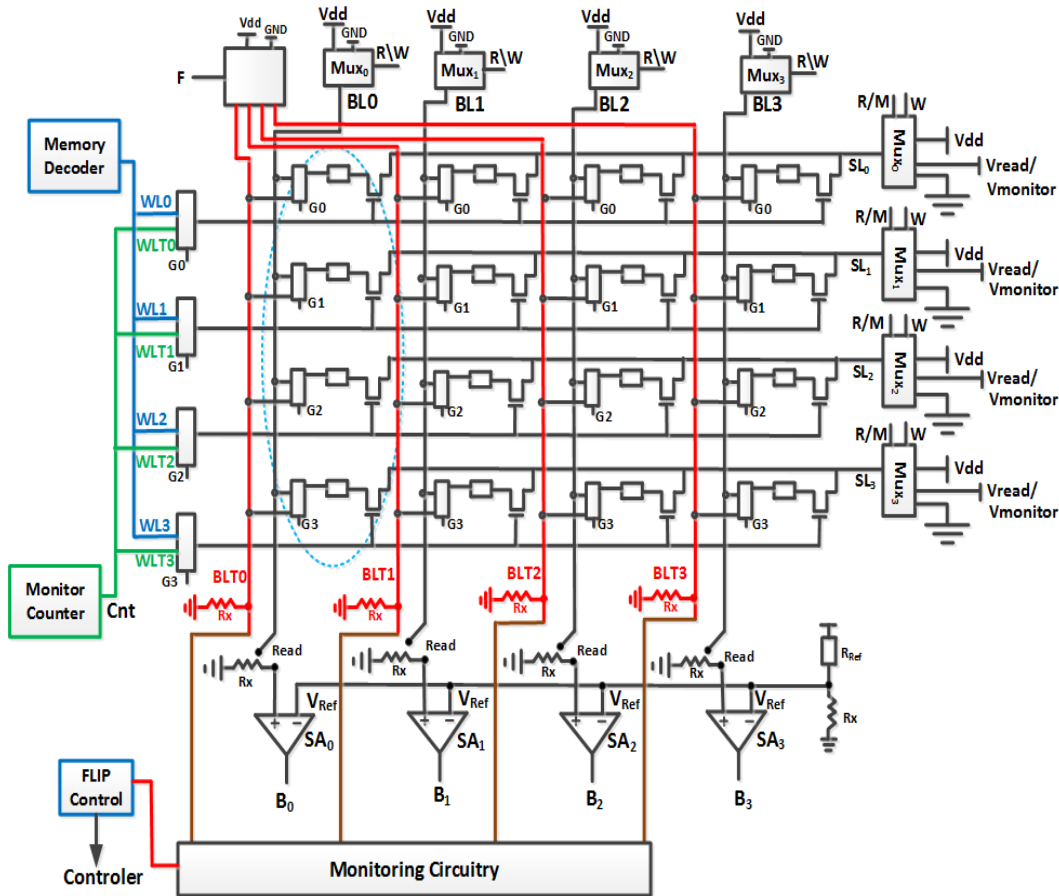


Figure 8.6: Monitor in crossbar column-by-column procedure

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

Again, similarly to the cell-by-cell approach, the monitoring process interferes with the normal operation of the memory in the SLs and WLs, and it would be needed to completely isolate the cells from all access lines inside the crossbar. This would require three multiplexer per cell including one to isolate the cell from bitline, one to isolate it from the wordline and one to separate the cell from select line. Therefore, it would not be appropriate for future memory structures where the objective is to reduce the size of each memory cell.

### 8.3.1.3 Row-by-Row Monitoring

To monitor the crossbar cells row-by-row, one specific row of the crossbar is selected, as it is depicted in Figure 8.7. Here, the monitoring procedure should also contain the measurement and flip phase. In the measurement phase, the LRS or HRS value is monitored and then, respectively in the flip phase the cell's data is reversed. The architecture includes the necessary units to perform the monitoring simultaneous to the normal memory operation.

In contrast with cell-by-cell and column-by-column that it was needed to isolate the cells from the SLs and WLs, in this monitoring strategy only the isolation from the bitline is necessary making overhead only in row multiplexers. Therefore, the monitoring and memory normal operation can be simultaneous and without any interference, what involves a more optimum time distribution and better system efficiency. This monitoring structure seems the best option for the monitoring procedure, and hereafter is chosen in this thesis.

## CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

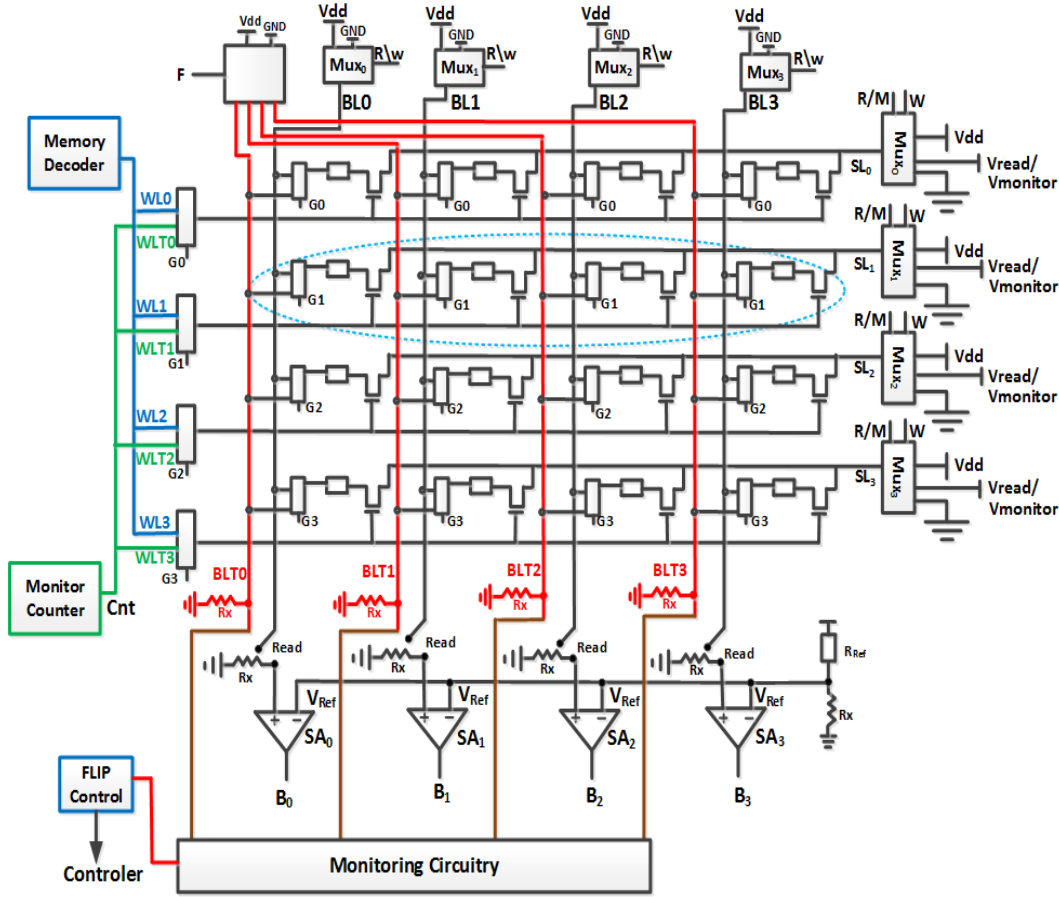


Figure 8.7: Monitor in crossbar row-by-row

### 8.3.1.4 Monitoring Approach Suggested in this Thesis

In Chapter 5, we assumed the existence of some idle cycles during the memory operation to copy the contents of the specific column under monitor. Then, each cell was monitored at that idle cycle, and its reliability status was evaluated. Here, and for implementing the monitoring mechanism, it is also considered that the crossbar memory can benefit from idle cycles during its operation.

Moreover, as it was discussed in the previous section, the row-by-row monitoring scenario in a crossbar seems to be the one with best granularity to implement the monitoring operation in each phase. This is due to the

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

fact that if the monitoring is row-by-row, then it is easier to control the monitoring flow rather than in the column-by-column approach. Also the area overhead can be reduced if the monitoring granularity is chosen to be at row level and not in column level, because less multiplexer would be needed.

This thesis will also benefit from this fact, and will implement the monitoring inside the crossbar row-by-row. As a consequence, the proposed monitoring approach in this thesis will monitor the reliability status of memristive cells and determine the functionality of memristive crossbar memory. An example of complete monitoring phases is explained in next section, where the monitoring will be utilized to direct the reconfiguration inside the crossbar.

### 8.4 Non-adaptive and Adaptive Crossbar Reconfiguring Strategies

Non-adaptive and Adaptive reconfigurations are the two approaches that this thesis proposed in Chapter 7 to extend the crossbar lifetime. So then, this section would introduce some implementation examples of these two concepts and analyzes them in brief.

#### 8.4.1 Non-adaptive Reconfiguring Strategy

Figure 8.8 displays a crossbar memory in non-adaptive reconfiguration approach. It is assumed that the active crossbar is  $m \times m = 4 \times 4$  and the global crossbar is  $N \times N = 8 \times 8$ . All circuits (i.e. read, write and monitor) are multiplexed to all  $m \times m$  structures. Therefore, the only added hardware are the monitoring circuit and the multiplexers.

Moreover, this architecture includes the required units not only for normal memory operation but also additionally consists of some circuits that perform the addressing, control and online monitoring in the crossbar. In the main array, starting from top left  $m \times m$  crossbar, only one  $m \times m$  crossbar is in active mode at time. Thus, one  $m \times m$  is used for storage purpose during the monitoring phase and the other  $m \times m$  units get functional, only when one cell of the previous  $m \times m$  stops working, by a fail caused by process variation

## CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

or endurance failure during the crossbar lifetime. Then, some multiplexers are utilized, which direct the read, write and monitor phase in the crossbar. These multiplexers are all managed by a controller, what makes possible the correct addressing. The read and write process are similar to the described method in the previous section, so in this section only the addressing and monitoring is described.

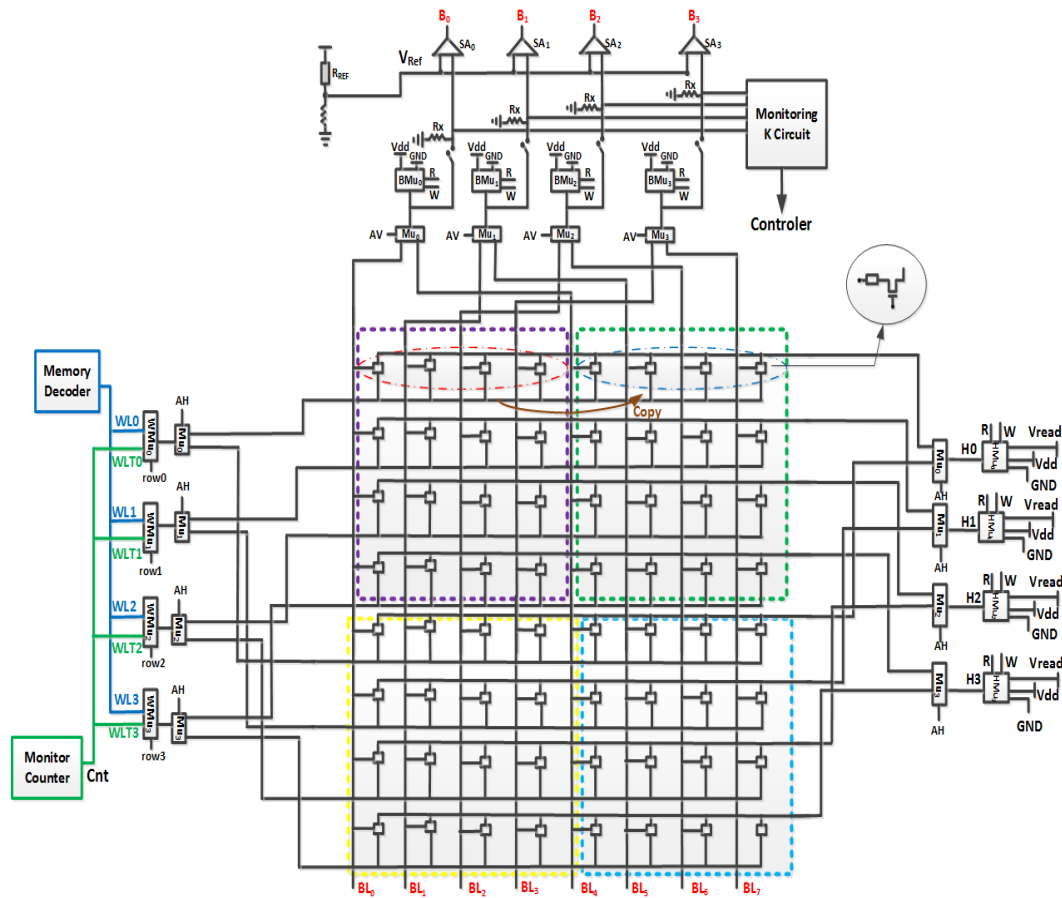


Figure 8.8: Non-adaptive reconfiguration architecture and implementation showing the monitoring units in the left and upper part

Then, it is assumed that the first  $m \times m$  is selected for memory operation inside the crossbar. Afterwards, to activate the proper rows and columns inside the crossbar, the corresponding select signals, as shown in Figure 8.9, can be applied to the multiplexers, and therefore, the correct  $N \times N$  section

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

would be under normal memory operation. In the next phases, when the selected mxm fails, the selected multiplexer signals will change correspondingly to choose the appropriate section of NxN as the proximate mxm framework for normal memory operation.

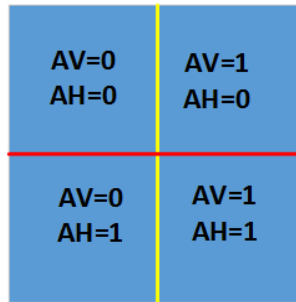


Figure 8.9: Addressing scheme inside the non-adaptive architecture for an example of 4 divided sub-sections of a global nxn crossbar

In this context, Figure 8.10 presents the designed algorithm to perform the monitoring for reconfiguration decision. Regarding this, the mxm memristive crossbar is functional and accessible for normal read and write operations. Then, when the memory enters into idle mode the monitoring is performed in such a way that it does not affect the memory cell's stored bits. Note that the monitoring will be executed inside the operational mxm, row by row and first, the stored bits in the WL that will undergo the monitoring are copied to the WL in adjunct mxm structure.

Next, the monitoring is performed in the corresponding WL, and finally the stored values are restored back to the original WL values. Figure 8.11 depicts the signals during the memory operation, observe that each step of copy, monitoring and restore contain sub-steps (reads and writes). In the copy, monitoring and restore steps, only the WL under monitor ( $WL_0$  in Figure 8.11) is active and other WLs are not active, and the BLs and SLs are activated correspondingly one by one for the read and write operations.

The main phase for monitoring includes a set of write and read operations, that first a '0' and then a '1' is written inside the cell and each time the memristor's resistance state is monitored through the monitoring circuits. Finally, the restore step recovers the original state of cells in the WL.



## CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

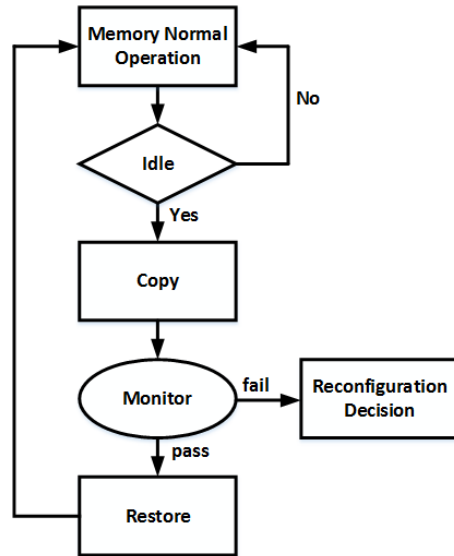


Figure 8.10: Monitoring flow in the non-adaptive

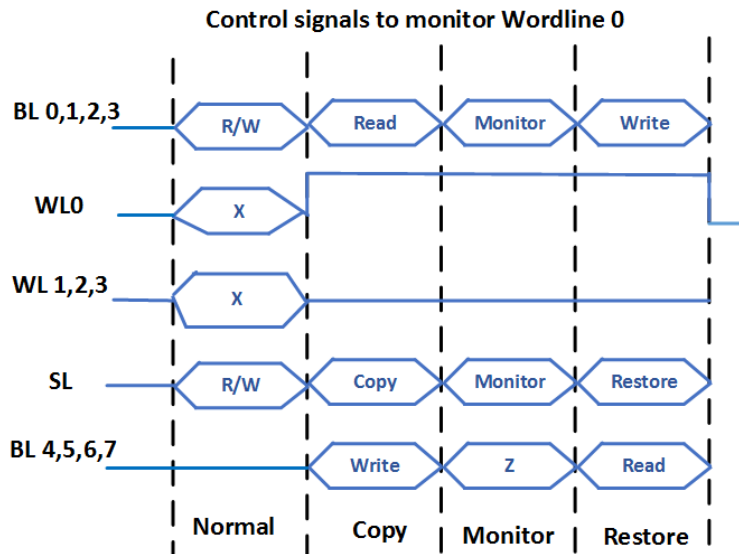


Figure 8.11: Monitoring diagrams in the memristive crossbar

### 8.4.2 Adaptive Reconfiguring Strategy

Chapter 7 highlighted the benefits of utilizing an adaptive reconfiguration in memristive crossbar. However, it would be interesting to design their architecture, making the crossbar matrix able to function adaptively by uti-

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

lizing the minimum possible added hardware and circuits to be more area efficient. Then, an implementation of adaptive reconfiguration technique in a memristive crossbar matrix is proposed in this thesis, which also includes a monitoring procedure.

For instance, Figure 8.12 presents the implementation of an adaptive shifting technique in a crossbar memory for  $m=4$  and  $N=8$ , just to simplify the example. In the main array, starting from top left  $m \times m$  crossbar, only one  $m \times m$  crossbar is in active mode at time. One  $m \times m$  is used for storage during the monitoring phase and the other cells get functional column by column, only when some part of the previous  $m \times m$  stops working, by a fail caused by process variation or endurance degradation during the crossbar lifetime. The addressing in the word-lines (at row level) is simply governed only by the row multiplexers like the non-adaptive approach, but since the adjustable shifting is through the columns, the column by column addressing is done by one addressing unit, which multiplexes the memory decoder to the appropriate set of  $m$  columns at a time.

Besides all these units, which are necessary for the memory operation, our adaptive approach also needs monitoring procedure and some circuits to monitor the memory cells alongside their lifetime to detect the ones which are close to fail (the cells with  $\frac{HRS}{LRS} = K$ ). This would help the system to reconfigure the working crossbar section on the fly before a fail could happen, and then a lifetime extension could be observed, as well. In this thesis, a monitoring technique based on analyzing the process variability and endurance degradation of the memristive cells during the memory lifetime is demonstrated. This monitoring approach can be applied to the cells, row-by-row and can evaluate the relation between HRS and LRS for a robust memory operation.

## CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

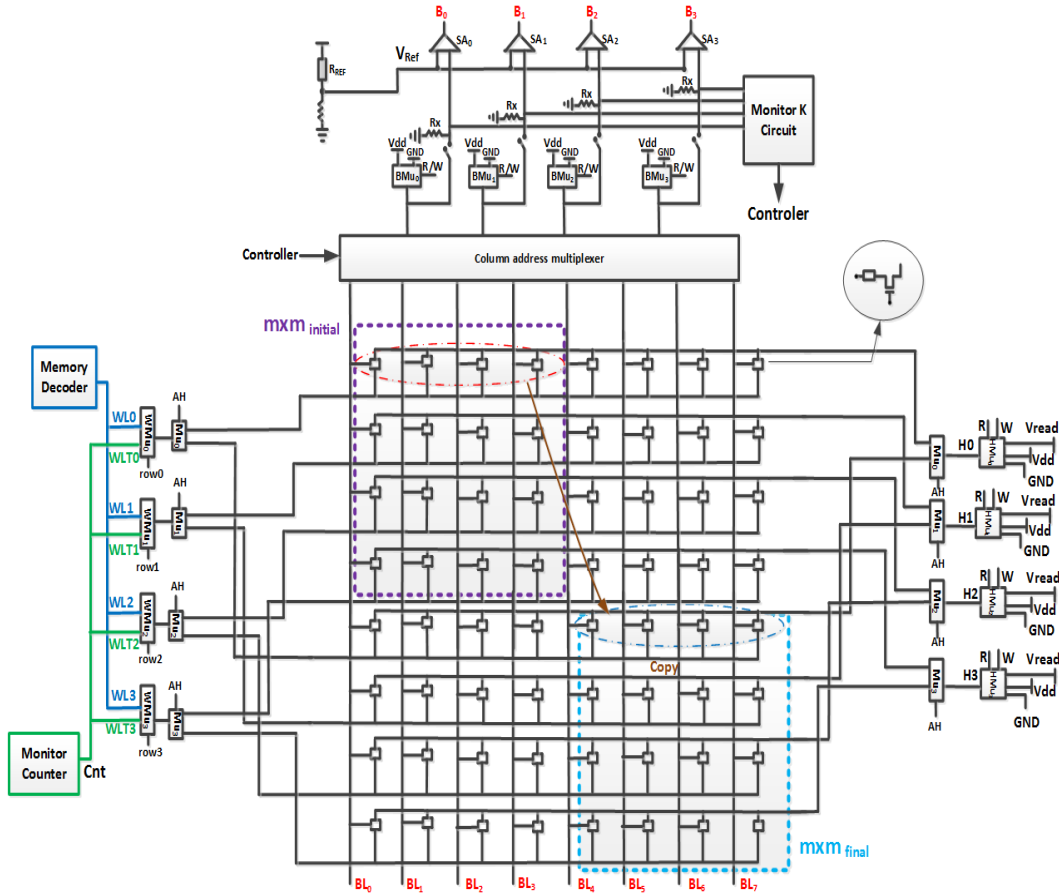


Figure 8.12: An architectural implementation of adaptive reconfiguration in crossbar architecture, where an  $mxm$  structure shifts according to location of its weakest or near-failing unit inside  $N \times N$  crossbar

Assuming the crossbar has just started after manufacturing, and the active  $mxm$  is the  $mxm_{initial}$ , the monitoring phase would start when the memory enters into idle stage. Here, we explain the monitoring of one word-line as an example ( $WL_0$  of  $mxm_{initial}$  in Figure 8.12):

1. The  $WL_0$  data is copied into the  $WL_0$  of  $mxm_{final}$ , which is used during the monitoring for storage purposes.
2. A '0' is written to all the cells in the  $WL_0$ , then, by monitoring the  $WL_0$  cells one by one through our monitoring circuits, the HRS value is monitored. This is done by performing an operation similar to read

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

and storing the memristor's current in a capacitor.

3. A '1' is written to all the cells in the  $WL_0$ , and similarly to previous step, the LRS value of memristive cells is monitored. The memristor's current would be now stored in another capacitor.
4. A voltage division is performed between the two voltages in the capacitors, which give the value close to  $\frac{HRS}{LRS}$ . If this value is an appropriate value in comparison with  $K$ , the corresponding word-line passes the monitoring phase.
5. Finally, step the data is restored to the  $WL_0$ . If the word-line does not pass the monitoring phase, in the adaptive approach the mxm shifts at column level according to the place of potential fail.

Observe that during all phases of monitoring (copy-measurement-restore) the data in the other rows is not altered. Such adaptive architecture and its monitoring scheme presented here can improve the reliability of the memristive crossbar and extend its lifetime.

### 8.4.3 The Monitoring Circuitry Implementation

The write, read and monitoring procedure in the crossbar can also be demonstrated with circuit simulations. In this sense, Figure 8.13 presents the details of one 1T1R cell as an example for write, and monitoring procedures (reading is also similar to monitoring but with sense amplifiers instead) in the memristive crossbar. This circuit is extracted from the proposed architecture in Figures 8.8 or 8.12.

This monitoring circuit is constructed with some switches ( $SW_M, SW_L, SW_H$ ), to control the correct passing current, two capacitors ( $C_{LRS}, C_{HRS}$ ) to store respecting voltages to LRS and HRS, a divisor to calculate the ratio between HRS and LRS, and a comparator to compare the division result with the design value of  $K$ .

CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

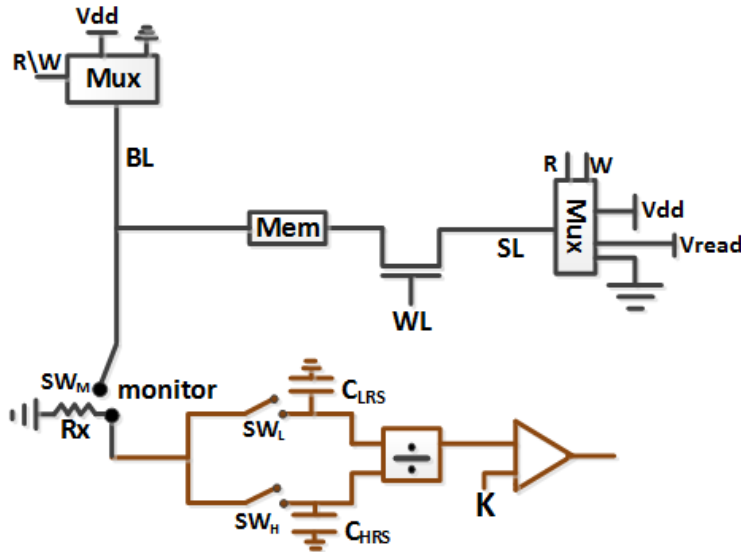


Figure 8.13: Monitoring circuit used in non-adaptive and adaptive crossbar architectures, it monitors the HRS and LRS values of the memristor

Such adaptive architecture and its monitoring scheme can improve the reliability of the memristive crossbar and extend its lifetime, though this reconfiguring technique induces an added extra area overhead to the memristive crossbar memory architecture.

Regarding this, Table 8.2 presents the hardware overhead of the adaptive approach in comparison with non-adaptive approach. Note that, the implemented architecture in this section can be easily modified for different realizations of  $m \times m$  and  $N \times N$ . Thus, the monitoring circuit would be the same as presented here, a global one, to monitor all memristive cells. Moreover, the number of word-line and source-line multiplexers will be always equal to  $2 \times m$  of 2-to-1 multiplexer units; only the adaptive multiplexing would become more complex as the number of  $N$  and  $m$  gets bigger.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

Table 8.2: Implementation Overhead of Adaptive Reconfiguring Approach

Section	Added Hardware
At word-lines	m Multiplexers of 2-to-1
At source-lines	m Multiplexers of 2-to-1
At Bit-lines	Adaptive multiplexing unit
Test units in crossbar	1 divisor, 1 comparator, 3 switches and 2 capacitors

### 8.4.3.1 Simulation

In order to perform the system simulations, this thesis uses the resistive switching (ReRAM) Verilog model designed by Stanford University [141] for the circuit simulations. This model is designed for bipolar metal oxide ReRAM devices based on conductive filament switching concept and has no limitations on the size of ReRAM cell.

We have considered all the default sizes in the model proposed by authors such as the cell size, which is equal to  $10 \times 10 \text{ nm}^2$ . Figure 8.14 shows the conductive filament concept in this model, where oxygen vacancies construct the filament between top and bottom electrodes [141].

The monitoring circuit shown in Figure 8.13 is simulated by HSPICE, and Figure 8.15 depicts the circuit simulation results obtained by HSPICE [109]. The simulation is transient and contains first writing a '0' inside the cell assuming LRS as an initial state. In order to write '0' inside the cell without interference with monitoring section we have considered some switches in the monitoring path. At this phase, the switches are as following  $SW_M$ =open to isolate the monitoring circuits and also  $SW_L, SW_H$ =open. At the moment of writing '0' the memristor current goes to almost zero, because the resistance state is switched from LRS to HRS and no current can flow inside the device.

Then, in the monitoring phase, we are interested to measure the value of HRS. This is done by applying appropriate voltage from the source-line and to close two of the switches in the circuit ( $SW_M$ =close,  $SW_H$ =close), while one switch is kept open  $SW_L$ =open for further monitoring steps. Figure 8.16

CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS

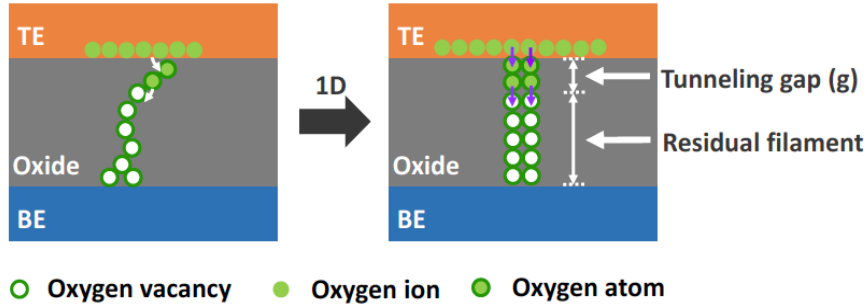


Figure 8.14: Illustration of ReRAM model [141]

shows in detail that the corresponding voltage to HRS is stored through our monitoring circuit and capacitor ( $C_{HRS}$ ), such that the  $VC_{HRS}$  is equal to the voltage divided between Rx and HRS.

Next the objective would be to measure the corresponding voltage according to LRS of the memristor. Therefore, it is needed to write '1' inside the cell. Again, the isolating switches should be open ( $SW_M, SW_L, SW_H = \text{open}$ ) and a transient HSPICE simulation is done. Figure 8.15 shows how the memristor current goes up to 300uA, due to its switching from HRS to LRS, in which more current can pass through the device. After writing '1' it is time to monitor the LRS status of the device by applying the appropriate voltage from the source-line and closing two switches such as ( $SW_M = \text{close}, SW_L = \text{close}$ ) and keeping one switch open ( $SW_H = \text{open}$ ) for proper monitoring. Then, accordingly the other capacitor in monitoring circuit ( $C_{LRS}$ ),  $VC_{LRS}$  holds the voltage divided between Rx and LRS.

After these, the monitoring phase continues with dividing these values and finding out their ratio that is  $\sim \frac{HRS}{LRS}$  (around 23 in this example for a fresh ReRAM cell in Stanford model). Note that, in this case the value of Rx is important and can have significant impact on the measurement if it is chosen big (Figure 8.16). In the phase of monitoring if the  $\frac{HRS}{LRS}$  value is acceptable (for instance in this specific example  $23 > K$ ) the cell is recognized as a healthy cell and if not it is identified as a weak, making the reconfiguration crucial in the future steps.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

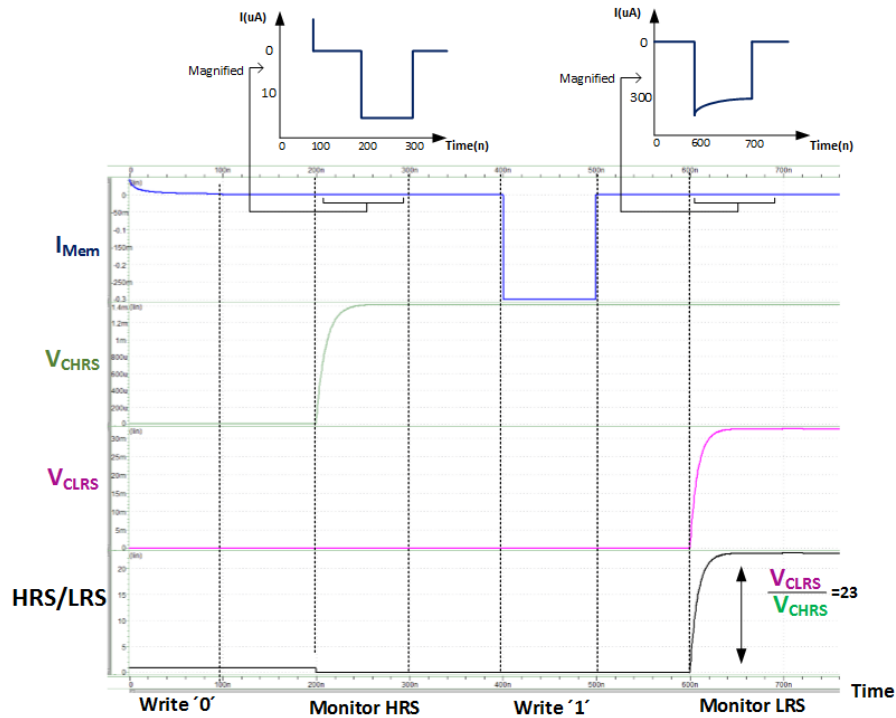


Figure 8.15: Circuit simulation waveforms in HSPICE showing the current waveforms at each monitoring phase

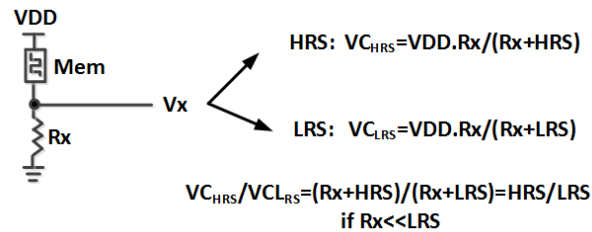


Figure 8.16: Monitoring circuit observations according to voltage divisions at the cell output

## 8.5 Summary and Conclusions

This Chapter of the thesis has analyzed the memristive memories and cross-bars from circuit perspectives. Therefore, its main contributions include proposing architectures to implement the reconfiguring approaches discussed and evaluated in the previous Chapter.



## **CHAPTER 8. MONITORING APPROACH IN MEMRISTIVE MEMORIES WITH RECONFIGURATION MECHANISMS**

---

Regarding this, first a conventional memristive memory structure and its read/write process for 1T1R cells was analyzed. Second, two example implementations for Non-adaptive and Adaptive reconfiguration strategies in crossbar memories was proposed at circuit level. Third, a novel monitoring approach was proposed to monitor the resistance ratio of the HRS and LRS in order to detect the weak cells inside the crossbar, this monitoring technique differs with existing approaches as it is an online mechanism.

Finally, the functionality and monitoring operation of the memristive cell by HSPICE simulation, verifies the proposed approach in this thesis. It was shown that the reliability of the memristive cell in the cell crossbar could be evaluated by the proposed monitoring circuit in this thesis and can be utilized in reconfiguring approaches.

The results of this Chapter has been published in Conference papers of European Conference on Circuit Theory and Design (ECCTD 2015) [142] and International Conference on Memristive Systems (MEMRISYS 2015) [131].



# 9

## Conclusion and Future work

Reliability of integrated circuits has gained a relevant importance with the trend of scaling in nano-electronic design paradigm. This fact imposes innovations at every level of design abstraction to enhance their reliability, in order to improve chip yield and extend the system lifetime.

Modern memories, occupying a significant area in current integrated circuits, are one of the key circuits to be considered in terms of being tolerant to reliability concerns such as process variation and aging. This thesis has considered two types of digital memories in order to be analyzed for reliability proposes: one kind, which is under production in advanced CMOS technologies, the SRAM array, and another one, which is emerging recently and regarded as a potential future embedded memory candidate, the memristor-based memories.

Both memory types benefit from existing variation and aging tolerant techniques at different design abstractions, where few of important such approaches are presented in Figure 9.1. This thesis has contributed in two of these abstractions by proposing new techniques and methodologies (highlighted in Figure 9.1) to enhance the memory reliability.

## Variation Tolerant Memory Design

SRAM Memories	Memristive Memories
<b>Device:</b> Improving device manufacturing process	<b>Device:</b> Improving material
<b>Circuit:</b> -Sizing -Body Bias -Voltage Scaling -8T and 10T, read and write assist circuits -On chip sensors	<b>Circuit:</b> -Parallel memristors -CRS based cells -Enhancing sense circuits -On chip sensors
<b>Architecture:</b> -Cache Resizing -ECC -Reconfiguration	<b>Architecture:</b> -ECC -Reconfiguration

Figure 9.1: A comparison of variation aware techniques in SRAM and memristive memories

Regarding this, next section briefly summarizes the main contribution of this thesis in the field of reliability aware memory design.

### 9.1 Summary of Contributions

This thesis has analyzed several challenges in the design and analysis of advanced reconfiguration mechanisms, on SRAM and memristive memories. The major contributions of this thesis include:

- Models of aging for both technologies CMOS and memristor-based. Modeling BTI aging in CMOS technologies, based on piecewise linear approximation of aging in long time periods. This model has been used for evaluating the lifetime of SRAM arrays in various reconfiguration techniques. Similarly and in the second part of thesis, the en-

## CHAPTER 9. CONCLUSION AND FUTURE WORK

---

durance degradation of memristive devices is modeled by linear equations. Then, this model is used to predict the memristive cell and crossbar lifetime in presence of process variation.

- New adaptive reconfiguration techniques for SRAM based on proactive concept. Proposing a process-variability aware proactive reconfiguration based on dynamic recovery allocation, which can extend the SRAM lifetime better than the existing proactive approach. This reconfiguring technique is evaluated analytically and then its benefit has been shown together with Monte-Carlo simulations. Moreover, the SRAM array memory architecture is modified to implement the proposed reconfiguring approach and the corresponding moderate area overhead is demonstrated.
- New adaptive reconfiguration techniques for memristor-based memories. Analyzing the reliability concerns in memristor-based memories and the importance of applying advanced reconfiguring techniques has been clarified. In this sense, two reconfiguration scenarios are presented and compared in this thesis work, where one is based on substituting the complete matrix and another is based on partial matrix substitution and dynamic shifting. The second approach has obtained better lifetime extension in comparison with the first one, demonstrated through Monte-carlo simulations. Furthermore, the memristive crossbar architecture is modified to implement the proposed reconfiguring techniques.
- Development of specific monitoring strategies and on-chip sensors to evaluate aging and process variability in SRAM and memristive memories. They can measure the variability and aging status of the memory cells and detect the weak cells in the array or crossbar. These on-chip sensors are integrated inside the memory architecture and circuit simulations results have demonstrated their monitoring applicability. In addition and in case of SRAM design a chip is fabricated with 350nm CMOS technology and the operation of on-chip monitoring sensor is verified.

## 9.2 Future work

This thesis work introduced the concept of adaptive proactive reconfiguration to extend the SRAM lifetime subjected to process variability and aging. Nevertheless, it is evident that this approach needs to be utilized in conjunction with other repair mechanisms such as reactive reconfiguration and ECC to improve the SRAM reliability. This combined utilization of repair techniques is left as a future work of this thesis. This task will present the reliability benefits of repair techniques together with each other and in front of various failure mechanisms such as transient or hard faults in SRAM memories.

Another future work for this thesis is the memory performance analysis at higher design abstractions such as in the micro-architectural level. It would be interesting to see the impacts of adaptive reconfiguration in cache performance, and to find flexible approaches in order to reduce the potential small performance losses.

Moreover, regarding the first part of thesis we could experimentally verify our monitoring concept to measure aging in SRAM cells. The similar experimental verification is left as a future work for the second part of thesis in context of memristive devices. The future work will be to first experimentally observe the switching behavior in a memristive device and the endurance and variability characteristics in a set of memristive devices. Then, it would be interesting to experimentally verify the introduced monitoring and measurement technique of thesis and track the aging of the memristive cells in an array.

Finally, it should be noted that one of the recent and emerging interesting topics in computer design field is the ability to mix the non-volatile concept with the existing volatile memory. Regarding this and since this thesis has analyzed both SRAM and one type of non-volatile memory another to be done future work is to analyze this hybrid structures such as the non-volatile SRAM (nvSRAM) [143]. It is a conventional SRAM cell modified with non-volatile devices such as ReRAM. The SRAM cell can be built with a conventional 6T SRAM cell and 2 ReRAM devices therefore is called 6T2R cell. It has a compact area and low power functionality, but though non-

volatile ReRAMs enable the store/restore without data loss when the power goes off, they have far less endurance than volatile memories such as SRAM. This motivates design of endurance-aware techniques in memory to improve their lifetime.

### 9.3 Publications

#### Journals:

- P.Pouyan, E.Amat, A.Rubio; "Adaptive Proactive Reconfiguration: A Technique for Process-Variability- and Aging-Aware SRAM Cache Design", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2014
- P.Pouyan, E.Amat, A.Rubio; "On-Chip Aging and Process-Variability Sensor in SRAM Memories", IEEE Sensors Journal, Under review.

#### Conferences:

- P.Pouyan, E.Amat, A.Rubio; "Insights to Memristive Memory Cell from a Reliability Perspective", to be appeared in MEMRISYS, Cyprus, 2015.
- P.Pouyan, E.Amat, A.Rubio; "Memristive Crossbar Design and Test in Non-adaptive Proactive Reconfiguring Scheme", ECCTD, Trondheim, Norway, 2015.
- P.Pouyan, E.Amat, A.Rubio; "Analysis and Design of an Adaptive Proactive Reconfiguration Approach for Memristive Crossbar Memories", NANOARCH, Boston, USA, 2015.
- P.Pouyan, E.Amat, A.Rubio; "Statistical Lifetime Analysis of Memristive Crossbar Matrix", DTIS, Naples, Italy, 2015.
- P.Pouyan, E.Amat, A.Rubio; "Reliability Challenges in Design of Memristive Memories", VARI, Palma de Mallorca, Spain, 2014.

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- P.Pouyan, E.Amat, E. Barajas, A.Rubio; "Impact of Proactive Reconfiguration Technique on Vmin and Lifetime of SRAM Caches", ISQED, San Jose, USA, 2014.
- P.Pouyan, E.Amat, F Moll, A.Rubio; "Design and Implementation of an Adaptive Proactive Reconfiguration technique in SRAM Caches", DATE, Grenoble, France, 2013.
- P.Pouyan, E.Amat, A.Rubio; "SRAM Lifetime Improvement Using Adaptive Proactive Reconfiguration", MIXDES, Warsaw, Poland, 2012.
- Peyman Pouyan, Esteve Amat, and Antonio Rubio, "Process-Variability Aware Proactive Reconfiguration Technique for Mitigating Aging Effects in Nano Scale SRAM Lifetime", VTS, Hawaii, USA, 2012.

### Workshops:

- P.Pouyan, E.Amat, A.Rubio; "Statistical Lifetime Analysis of Memristive Crossbar", Unreliability Workshop WP7 DATE, Grenoble, France, 2015.
- P.Pouyan, E.Amat, F Moll, A.Rubio; "Reconfiguration Proactive Techniques for Process and Aging Variation Aware Cache Design", VARI, Karlsruhe, Germany, 2013.
- P.Pouyan, E.Amat, A.Rubio; "Proactive Reconfiguration, a methodology for extending SRAM lifetime", VAMM Workshop, DATE Poster session, Dresden, Germany, 2012.



# Bibliography

- [1] “<http://www.itrs.net/>,” *International Technology Roadmap for Semiconductors*, 2013. 1, 2, 5, 11, 17, 21, 23, 24, 135
- [2] M.Haselman and S.Hauck, “The Future of Integrated Circuits: A Survey of Nanoelectronics,” *Proceedings of IEEE*, vol. 98, no. 1, pp. 11–38, November 2009. 1, 3
- [3] C. Mack, “Fifty Years of Moore’s law,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 24, no. 2, pp. 202–207, January 2011. 2, 4
- [4] M. Aoulaiche, *Bias Temperature Instabilities in MOSFETs with High-K Dielectrics and Metal Gates*, *PhD Thesis*. Imec, 2009. 2, 7, 64, 65
- [5] T. S.E, S. Guangyu, C. Y. Sung, and N. Toshikazu, “Uniaxial-Process-Induced Strained-Si:Extending the CMOS Roadmap,” *IEEE Transactions on Electron Devices*, vol. 53, no. 5, pp. 1010–1020, May 2006. 2
- [6] J. Hicks, D. Bergstrom, M. Hattendorf, J. Jopling, C. Prasad, and J. Wiedemer, “45nm Transistor Reliability,” *Intel Technology Journal*, vol. 12, no. 2, pp. 131–142, June 2008. 2, 59
- [7] “<http://www.st.com/>,” *Learn more about FD-SOI*, 2015. 2
- [8] D. Hisamoto, W. chin Lee, J. Kedzierski, H. Takeuchi, K. Asano, C. Kuo, E. Anderson, T. jae King, J. Bokor, and C. Hu, “FinFET-A

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- Self-Aligned Double-Gate MOSFET Scalable to 20 nm,” *IEEE Transactions on Electron Devices*, vol. 47, no. 12, pp. 2320–2325, December 2000. 2
- [9] S. Xin, L. Qiang, V. Moroz, H. Takeuchi, G. Gebara, J. Wetzel, I. Shuji, S. Changhwan, and K. L. Tsu-Jae, “Tri-Gate Bulk MOSFET Design for CMOS Scaling to the End of the Roadmap,” *IEEE Electron Devices Letters*, vol. 29, no. 5, pp. 491–493, May 2008. 2
- [10] A. M. Ionescu and H. Riel, “Tunnel field-effect transistors as energy-efficient electronic switches,” *Nature*, vol. 479, pp. 329–337, November 2011. 3
- [11] C. G. Almudever, *Variability and Reliability Analysis of Carbon Nanotube Technology in the Presence of Manufacturing Imperfections*, PhD Thesis. Universitat Politecnica de Catalunya, 2014. 3
- [12] S. Luryi, J. Xu, and Zaslavsky, *Nanowires: Technology, Physics and Perspectives*. Wiley, 2010. 3
- [13] P. Mazumder, S. Kang, and R. Waser, “Memristors: Devices, Models, and Applications,” *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1911–1919, June 2012. 3, 136
- [14] K. Zhang, *Embedded Memories for Nano-Scale VLSIs*. Springer, 2009. xvii, 4, 25, 26, 27, 28, 31
- [15] K. Itoh, *VLSI Memory Chip Design*. Springer, 2001. 5, 17, 18
- [16] H.-S. Wong, L. Heng-Yuan, Y. Shimeng, C. Yu-Sheng, W. Yi, C. Pang-Shiu, L. Byoungil, F. Chen, and T. Ming-Jinn, “Metal Oxide RRAM,” *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, May 2012. 5, 23, 138
- [17] S. Ghosh and K. Roy, “Parameter Variation Tolerance and Error Resiliency: New Design Paradigm for the Nanoscale Era,” *Proceedings of IEEE*, vol. 98, no. 10, pp. 1718–1751, October 2010. 6, 7, 8, 37, 51, 58, 112

## BIBLIOGRAPHY

---

- [18] X. Wang, A. Brown, B. Cheng, and A. Asenov, “Statistical Variability and Reliability in Nanoscale FinFETs,” in *IEEE International Electron Devices Meeting (IEDM)*, 2011. xviii, 7, 59, 61
- [19] G. Gielen, P. D. Wit, E. Maricau, J. Loeckx, J. Martin, B. Kaczer, G. Groeseneken, R. Rodriguez, and M. Nafria, “Emerging yield and reliability challenges in nanometer CMOS technologies,” in *International Conference Design Automation and Test in Europe (DATE)*, 2008. 7
- [20] T. Grasser, *Bias Temperature Instability for Devices and Circuits*. Springer, 2014. 7
- [21] E. Amat, T. Kauerauf, R. Degraeve, R. Rodriguez, M. Nafria, X. Aymerich, and G. Groeseneken, “Channel hot-carrier degradation in pMOS and nMOS short channel transistors with high-k dielectric stack,” *Microelectronic Engineering*, vol. 87, no. 1, pp. 47–50, January 2010. 7, 70
- [22] T. Kauerauf, *Degradation and breakdown of MOS gate stacks with high permittivity dielectrics, PhD Thesis*. IMEC, 2007. 7, 71
- [23] S. Khan, I. Agbo, S. Hamdioui, H. Kukner, B. Kaczer, P. Raghavan, and F. Catthoor, “Bias Temperature Instability analysis of FinFET based SRAM cells,” in *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2014. xix, 7, 69, 75, 76
- [24] B. Chen, Y. Lu, B. Gao, Y. H. Fu, F. F. Zhang, P. Huang, Y. S. Chen, L. F. Liu, X. Y. Liu, J. F. Kang, Y. Y. Wang, Z. Fang, H. Y. Yu, X. Li, X. P. Wang, N. Singh, G. Q. Lo, and D. L. Kwong, “Physical mechanisms of endurance degradation in TMO-RRAM,” in *IEEE International Electron Devices Meeting (IEDM)*, 2011. xxi, 8, 139, 140, 145, 152, 153
- [25] J. Rajendran, R. Karri, and G. Rose, “Improving Tolerance to Variations in Memristor-Based Applications Using Parallel Memristors,” *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 733–746, February 2014. 8

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- [26] P. Royer, *Design and Simulation of deep nanometer SRAM cells under energy, mismatch, and radiation constrains, PhD Thesis*. University Polytechnic Madrid, 2015. 8
- [27] M. Horiguchi and K. Itoh, *Nanoscale Memory Repair*. Springer, 2011. xviii, 8, 11, 12, 36, 37, 40, 41, 42, 43, 45, 47
- [28] S. Sapatnekar, “Overcoming Variations in Nanometer-Scale Technologies,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 1, pp. 5–18, May 2011. 8, 57
- [29] L.-T. Wang and C. E. Stroud, *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer, 2010. 8
- [30] M. Stainisavljevic, A. Schmid, and Y. Leblebici, *Reliability of Nanoscale Circuits and Systems*. Springer, 2011. 9, 10
- [31] N. Aymerich, *Variability-aware Architectures based on Hardware Redundancy for Nanoscale Reliable Computation, PhD Thesis*. University Polytechnic Catalunya, 2013. xvii, 10
- [32] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, “A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology,” *Science*, vol. 280, no. 5370, pp. 1716–1721, June 1998. 11
- [33] J. Shin, V. Zyuban, P. Bose, and T. Pinkston, “A Proactive Wearout Recovery Approach for Exploiting Microarchitectural Redundancy to Extend Cache SRAM Lifetime,” in *International Symposium on Computer Architecture (ISCA)*, 2008. xviii, 11, 12, 52, 54, 55, 79, 81, 83, 88
- [34] S. ADEE, “Transistors Go Vertical,” *IEEE Spectrum*, November 2007. 18
- [35] A. Rubio, “TRAMS Deliveral SUB 16 Nm Technology FinFET Transistors Including Statistical Variability and Statistical Reliability, European Project,” November 2010. xvii, xviii, 18, 20, 32, 33, 61, 62

## BIBLIOGRAPHY

---

- [36] Z. Jaksic, *Cache Memory Design in the FinFET Era, PhD Thesis*. University Polytechnic Catalunya, 2015. 20, 30
- [37] S. Gupta and K. Roy, “Device-Circuit Co-Optimization for Robust Design of FinFET-Based SRAMs,” *IEEE Design and Test*, vol. 30, no. 6, pp. 29–39, June 2013. 20
- [38] T. Ludwig, I. Aller, V. Gernhoefer, J. Keinert, E. Nowak, R. Joshi, A. Mueller, and S. Tomaschko. 20
- [39] L.Chua, “Memristor-The missing circuit element,” *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, September 1971. 21
- [40] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams. xvii, 21, 22, 132, 137
- [41] L.Chua, “Resistance switching memories are memristors,” *Applied Physics*, vol. 102, no. 4, pp. 765–783, March 2011. 22
- [42] L. Wang, C. Yang, J. Wen, S. Gai, and Y. Peng, “Overview of emerging memristor families from resistive memristor to spintronic memristor,” *Journal of Materials Science Materials in Electronics*, vol. 26, no. 7, pp. 4618–4628, June 2015. 22, 23
- [43] A. Pavlov, *CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies: Process-Aware SRAM Design and Test*. Springer, 2008. xvii, 26, 32, 47
- [44] K. Noda, K. Matsui, K. Takeda, and N. Nakamura, “A loadless CMOS four-transistor SRAM cell in a 0.18-um logic technology,” *IEEE Transactions on Electron Devices*, vol. 48, no. 12, pp. 2851–2855, December 2001. 27
- [45] W. Luk *et al.*, “A 3-Transistor DRAM Cell with Gated Diode for Enhanced Speed and Retention Time,” in *VLSI Symposium Digest of Technical Papers*, 2006. 29, 30

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- [46] E. Amat, A. Calomarde, F. Moll, R. Canal, and A. Rubio, “Feasibility of the embedded DRAM cells implementation with FinFET devices,” *IEEE Transactions on Computers*, vol. pp, no. 99, pp. 1–12, December 2014. 30
- [47] J. J. Yang, D. B. Strukov, and D. R. Stewart, “Memristive devices for computing,” *Nature Nanotechnology*, vol. 8, pp. 13–24, August 2012. xvii, 33, 34
- [48] I. Vourkas and G. Sirakoulis, “A Novel Design and Modeling Paradigm for Memristor-Based Crossbar Circuits,” *IEEE Transactions on Nanotechnology*, vol. 11, no. 6, pp. 1151–1159, September 2012. 33
- [49] I. Vourkas, D. Stathis, G. Sirakoulis, and S. Hamdioui, “Alternative Architectures Toward Reliable Memristive Crossbar Memories,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–12, January 2015. 34
- [50] H. Manem, J. Rajendran, and G. S. Rose, “Design Considerations for Multilevel CMOS/Nano Memristive Memory,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 8, no. 1, February 2012. xvii, 34, 35, 146, 176
- [51] M. Gholipour and N. Masoumi, “Design investigation of nanoelectronic circuits using crossbar-based nanoarchitectures,” *Microelectronics Journal*, vol. 44, no. 3, pp. 190–200, March 2013. xvii, 34, 35
- [52] M. AbuRahma and M. Anis, *Nanometer Vriation-Tolerant SRAM*. Springer, 2011. xviii, 37, 59
- [53] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, “Modeling the effect of technology trends on the soft error rate of combinational logic,” in *International Conference on Dependable Systems and Networks*, 2002. xviii, 36, 37
- [54] M. Nicolaidis, *Soft Errors in Modern Electronic Systems*. Springer, 2010. 38, 39

## BIBLIOGRAPHY

---

- [55] R. Vattikonda, W. Wenping, and C. Yu, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Design Automation Conference (DAC)*, 2006. 39
- [56] T. Mano, J. Yamada, J. Inoue, and S. Nakajima, "Circuit techniques for a VLSI memory," *IEEE Journal of Solid-State Circuits*, vol. 18, no. 5, pp. 463–470, October 1983. 40
- [57] H. Kalter, C. Stapper, J. Barth, J. DiLorenzo, C. Drake, J. Fifield, G. Kelley, S. Lewis, W. van der Hoeven, and J. Yankosky, "A 50-ns 16-Mb DRAM with a 10-ns data rate and on-chip ECC," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 5, pp. 1118–1128, October 1990. 40
- [58] K. Arimoto, K. Fujishima, Y. Matsuda, M. Tsukude, T. Oishi, W. Wakamiya, S. ichi Satoh, M. Yamada, and T. Nakano, "A 60-ns 3.3-V-only 16-Mbit DRAM with multipurpose register," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1184–1190, October 1989. 40
- [59] K. Ganapathy, A. Singh, and D. Pradhan, "Yield optimization in large RAM's with hierarchical redundancy," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 9, pp. 1259–1264, September 1991. 41
- [60] A. Chen, "Redundancy in LSI memory array," *IEEE Journal of Solid-State Circuits*, vol. 4, no. 5, pp. 291–293, October 1969. 41
- [61] K. Ilyoung, Y. Zorian, G. Komoriya, H. Pham, F. Higgins, and J. Lewandowski, "Built in Self Repair for Embedded High Density SRAM," in *International Test Conference Proceedings*, 1998. xviii, 41, 47, 48, 80
- [62] S.-Y. Kuo and W. Fuchs, "Efficient Spare Allocation for Reconfigurable Arrays," *IEEE Design and Test of Computers*, vol. 4, no. 1, pp. 24–31, February 1987. 42
- [63] M. Horiguchi, "Redundancy techniques for high-density DRAMs," in *IEEE International Conference on Innovative Systems in Silicon*, 1997. 42

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- [64] S. Kumar, C. Kim, and S. Sapatnekar, "Impact of NBTI on SRAM read stability and design for reliability," in *International Symposium on Quality Electronic Design*, 2006. 49
- [65] S. Mukhopadhyay, C. Qikai, and K. Roy, "Memories in Scaled technologies: A Review of Process Induced Failures, Test methodologies, and Fault Tolerance," in *IEEE Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, 2007. 50
- [66] H. Mostafa, M. Anis, and M. Elmasry, "Adaptive Body Bias for Reducing the Impacts of NBTI and Process Variations on 6T SRAM Cells," *IEEE Transactions on Circuits and Systems I*, vol. 58, no. 12, pp. 2859–2871, July 2011. 51
- [67] L. Li, Y. Zhang, and J. Yang, "Proactive recovery for BTI in high-k SRAM cells," in *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2011. 53, 79
- [68] Y. Li, Y. Shao-Ming, H. Jiunn-Ren, and Y. Fu-Liang, "Discrete Dopant Fluctuations in 20-nm/15-nm-Gate Planar CMOS," *IEEE Transactions on Electron Device*, vol. 55, no. 6, pp. 1449–1455, June 2008. xviii, 58
- [69] A. Srivastava and D. Sylvester, *Statistical Analysis and Optimization for VLSI: Timing and Power*. Springer, 2005. 59
- [70] A. Asenov, "Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 um MOSFET's: A 3-D "atomistic" simulation study," *IEEE Transactions on Electron Devices*, vol. 45, no. 12, pp. 2505–2513, December 1998. xviii, 59, 60
- [71] D. Lu, L. Chung-Hsun, A. Niknejad, and H. Chenming, "Compact Modeling of Variation in FinFET SRAM Cells," *IEEE Design and Test of Computers*, vol. 27, no. 2, pp. 44–50, March 2010. 59
- [72] C. A. Mack, *Field Guide to Optical Lithography*. SPIE, 2006. xviii, 60, 61



- [73] W. Xingsheng, A. Brown, N. Idris, S. Markov, G. Roy, and A. Asenov, "Statistical Threshold-Voltage Variability in Scaled Decananometer Bulk HKMG MOSFETs: A Full-Scale 3-D Simulation Scaling Study," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2293–2301, May 2011. 61
- [74] M. Cho, M. Aoulaiche, R. Degraeve, B. Kaczer, J. Franco, T. Kauerauf, P. Roussel, L. Ragnarsson, J. Tseng, T. Hoffmann, and G. Groeseneken, "Positive and negative bias temperature instability on sub-nanometer eot high-K MOSFETs," in *IEEE International Reliability Physics Symposium (IRPS)*, 2010. 64
- [75] K. Kunhyuk, H. Kufuoglu, K. Roy, and M. Alam, "Impact of Negative-Bias Temperature Instability in Nanoscale SRAM Array: Modeling and Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 10, pp. 1770–1781, October 2007. xviii, 64, 65
- [76] T. Grasser, W. Gos, V. Sverdlov, and B. Kaczer, "The Universality of NBTI Relaxation and its Implications for Modeling and Characterization," in *IEEE International Reliability Physics Symposium (IRPS)*, 2007. 65, 72
- [77] W. Wenping, Y. Shengqi, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and C. Yu, "The impact of nbtI on the performance of combinational and sequential circuits." 65
- [78] R. Vattikonda, W. Wenping, and C. Yu, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," in *ACM/IEEE Design Automation Conference (DAC)*, 2006. 65
- [79] A. Bansal, R. Rao, J.-J. Kim, S. Zafar, J. H. Stathis, and C.-T. Chuanga, "Impacts of NBTI and PBTI on SRAM static/dynamic noise margins and cell failure probability," *Microelectronics Reliability*, vol. 49, no. 6, pp. 642–649, October 2008. xviii, xix, 65, 72, 74, 75, 76, 100, 124

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- [80] S. Zafar, Y. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, and M. Chudzik, "A Comparative Study of NBTI and PBTI (Charge Trapping) in SiO<sub>2</sub>/HfO<sub>2</sub> Stacks with FUSI, TiN, Re Gates," in *Symposium on VLSI Technology Digest of Technical Papers*, 2006. 65, 72
- [81] K. Kunhyuk, S. Gangwal, P. S. Phill, and K. Roy, "NBTI induced performance degradation in logic and memory circuits: how effectively can we approach a reliability solution?" in *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2008. 65, 66
- [82] S. Ramey, C. Prasad, M. Agostinelli, P. Sangwoo, S. Walstra, S. Gupta, and J. Hicks, "Frequency and recovery effects in high-k BTI degradation," in *IEEE International Reliability Physics Symposium (IRPS)*, 2009. xviii, 66
- [83] T. Grasser, B. Kaczer, P. Hehenberger, W. Gos, R. O'Connor, H. Reisinger, W. Gustin, and C. Schunder, "Simultaneous Extraction of Recoverable and Permanent Components Contributing to Bias-Temperature Instability," in *IEEE International Electron Devices Meeting (IEDM)*, 2007. xviii, xix, 66, 67, 73, 90
- [84] H. Kukner, P. Weckx, J. Franco, M. Toledano-Luque, C. Moonju, and B. Kaczer, "Scaling of BTI reliability in presence of time-zero variability," in *IEEE International Reliability Physics Symposium (IRPS)*, 2014. xix, 68
- [85] L. K. Taek, C. Minhyeok, L. Nae-In, A. Patel, P. Junekyun, and P. Jongwoo, "Technology scaling on High-K and Metal-Gate FinFET BTI reliability," in *IEEE International Reliability Physics Symposium (IRPS)*, 2013. xix, 68, 69
- [86] S. Ramey, J. Hicks, L. Liyanage, and S. Novak, "BTI recovery in 22nm tri-gate technology," in *IEEE International Reliability Physics Symposium (IRPS)*, 2014. xix, 68, 69

## BIBLIOGRAPHY

---

- [87] T. Hiramoto, M. Suzuki, X. Song, K. Shimizu, T. Saraya, A. Nishida, T. Tsunomura, S. Kamohara, K. Takeuchi, and T. Mogami, "Direct Measurement of Correlation Between SRAM Noise Margin and Individual Cell Transistor Variability by Using Device Matrix Array," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2249–2256, May 2011. xix, 74, 75
- [88] P. Pouyan, E. Amat, and A. Rubio, "Process-Variability Aware Proactive Reconfiguration Technique for Mitigating Aging Effects in Nano Scale SRAM Lifetime," in *IEEE VLSI Test Symposium (VTS)*, 2012. 77, 110
- [89] "Matlab: [4] [www.mathworks.com](http://www.mathworks.com)." 80
- [90] A. Krishnan, V. Reddy, D. Aldrich, J. Raval, K. Christensen, J. Rosal, C. O'Brien, R. Khamankar, A. Marshall, W.K.Loh, R. McKee, and S. Krishnan, "SRAM Cell Static Noise Margin and VMIN Sensitivity to Transistor Degradation," in *IEEE International Electron Devices Meeting (IEDM)*, 2006. 100
- [91] A. Carlson, "Mechanism of Increase in SRAM Vmin Due to Negative-Bias Temperature Instability," *IEEE Transactions on Device and Materials Reliability*, vol. 7, no. 3, pp. 1530–4388, September 2007. 100
- [92] M. Ball, J. Rosal, R. McKee, W. Loh, and T. Anderson, "A Screening Methodology for VMIN Drift in SRAM Arrays with Application to Sub-65nm Nodes," in *IEEE International Electron Devices Meeting (IEDM)*, 2006. 100
- [93] K. Lingkai, , L. Yue, and E. Alon, "A multi-GHz area-efficient comparator with dynamic offset cancellation," in *IEEE Custom Integrated Circuits Conference (CICC)*, 2011. 108, 109
- [94] S. Barasinski, L. Camus, and S. Clerc, "A 45nm single power supply SRAM supporting low voltage operation down to 0.6V," in *European Solid-State Circuits Conference (ESSCIRC)*, 2008. 108, 109

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- [95] P. Pouyan, E. Amat, and A. Rubio, “SRAM Lifetime Improvement Using Adaptive Proactive Reconfiguration,” in *International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES)*, 2012. 110
- [96] P. Pouyan, E. Amat, F. Moll, and A. Rubio, “Design and Implementation of an Adaptive Proactive Reconfiguration technique in SRAM Caches,” in *International Conference Design Automation and Test in Europe (DATE)*, 2013. 110, 130
- [97] P. Pouyan, E. Amat, and A. Rubio, “Impact of Proactive Reconfiguration Technique on Vmin and Lifetime of SRAM Caches,” in *International Symposium on Quality Electronic Design (ISQED)*, 2014. 110, 130
- [98] P. Pouyan, E. Amat, and A. Rubio, “Adaptive Proactive Reconfiguration: A Technique for Process Variability and Aging Aware SRAM Cache Design,” *IEEE Transactions on VLSI*, vol. 23, no. 9, pp. 1951–1955, September 2015. 110, 130
- [99] J. Keane and C. H. Kim, “Transistor Aging,” *IEEE Spectrum*, April 2011. 111
- [100] K. Tae-Hyoung, R. Persaud, and C. Kim, “Silicon Odometer: An On-Chip Reliability Monitor for Measuring Frequency Degradation of Digital Circuits,” in *IEEE Symposium on VLSI Circuits*, 2007. 111
- [101] J. Keane, W. Xiaofei, D. Persaud, and C. Kim, “An All-In-One Silicon Odometer for Separately Monitoring HCI, BTI, and TDDB,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 817–829, March 2010. 111
- [102] K. Kunhyuk, M. Alam, and K. Roy, “Characterization of NBTI induced temporal performance degradation in nano-scale SRAM array using IDDQ,” in *IEEE International Test Conference (ITC)*, 2007. xx, 112, 113, 114

## BIBLIOGRAPHY

---

- [103] F. Ahmed and L. Milor, “Reliable cache design with on-chip monitoring of NBTI degradation in SRAM cells using BIST,” in *VLSI Test Symposium (VTS)*, 2010. xx, 112, 114, 115
- [104] Q. Zhenyu, W. Jiajing, A. Cabe, S. Wooters, T. Blalock, B. Calhoun, and M. Stan, “SRAM-based NBTI/PBTI sensor system design,” in *IEEE Design Automation Conference (DAC)*, 2010. 112, 116
- [105] A. Ceratti, T. Copetti, L. Bolzani, and F. Vargas, “On-chip aging sensor to monitor NBTI effect in nano-scale SRAM, booktitle = IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS),” 2012. xxi, 112, 116
- [106] Y. Teng, K. Doyun, P. Kinget, and S. Mingoo, “In-situ techniques for in-field sensing of NBTI degradation in an SRAM register file,” in *IEEE International Solid- State Circuits Conference (ISSCC)*, 2015. xxi, 112, 116, 117
- [107] T. Siddiqua, S. Gurumurthi, and M. Stan, “Modeling and analyzing NBTI in the presence of Process Variation,” in *International Symposium on Quality Electronic Design (ISQED)*, 2011. 112
- [108] “Predictive technology models: online: <http://ptm.asu.edu>, 2011.” 119
- [109] “Hspice: online: <http://www.synopsys.com>.” 119, 194
- [110] “Cadence: online: <http://www.cadence.com>.” 124
- [111] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, “Switching dynamics in titanium dioxide memristive devices,” *Journal of Applied Physics*, vol. 106, no. 7, pp. 074 508–074 516, November 2009. 132, 134
- [112] D. Ielmini, “Modeling the Universal Set/Reset Characteristics of Bipolar RRAM by Field- and Temperature-Driven Filament Growth,” *IEEE Transactions on Electron Devices*, vol. 58, no. 12, pp. 4309–4317, October 2011. 132, 135

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- [113] G. Bersuker, D. Gilmer, D. Veksler, J. Yum, H. Park, S. Lian, L. Vandelli, A. Padovani, M. Nafria, W. Taylor, P. Kirsch, and R. Jammy, “Metal oxide RRAM switching mechanism based on conductive filament microscopic properties,” in *IEEE International Electron Devices Meeting (IEDM)*, 2010. 132
- [114] E. Linn, A. Siemon, R. Waser, and S. Menzel, “Applicability of Well-Established Memristive Models for Simulations of Resistive Switching Devices,” *IEEE Transactions on Circuits and Systems I*, vol. 61, no. 8, pp. 2402–2410, July 2014. 132
- [115] I. Vourkas, A. Batsos, and G. Sirakoulis, “SPICE modeling of nonlinear memristive behavior,” *International Journal of Circuit Theory and Applications*, vol. 43, no. 5, pp. 553–565, October 2013. 132
- [116] D. Biolek and V. Biolkova, “SPICE Model of Memristor with Nonlinear Dopant Drift,” in *Radioengineering Conference*, 2009. 133
- [117] A. Fantini, L. Goux, R. Degraeve, D. Wouters, N. Raghavan, G. Kar, A. Belmonte, Y.-Y. Chen, B. Govoreanu, and M. Jurczak, “Intrinsic switching variability in HfO<sub>2</sub> RRAM,” in *IEEE International Memory Workshop (IMW)*, 2013. 136
- [118] D. Niu, Y. Chen, C. Xu, and Y. Xie, “Impact of process variations on emerging memristor,” in *IEEE Design Automation Conference (DAC)*, 2010. 137
- [119] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, and D. Ielmini, “Understanding switching variability and random telegraph noise in resistive RAM,” in *IEEE International Electron Devices Meeting (IEDM)*, 2013. 138
- [120] S. Deora, G. Bersuker, K. Matthews, D. Gilmer, and P. Kirsch, “AC Variability and Endurance Measurement Technique for Resistive Switching Memories,” *IEEE Transactions on Device and Materials Reliability*, vol. 14, no. 1, pp. 300–303, November 2013. 138

## BIBLIOGRAPHY

---

- [121] C. An and L. Ming-Ren, “Variability of resistive switching memories and its impact on crossbar array performance,” in *IEEE International Reliability Physics Symposium (IRPS)*, 2011. 138
- [122] Y. Lu, B. Chen, B. Gao, Z. Fang, Y. Fu, J. Yang, L. Liu, X. Liu, H. Yu, and J. Kang, “Improvement of endurance degradation for oxide based resistive switching memory devices correlated with oxygen vacancy accumulation effect,” in *IEEE International Reliability Physics Symposium (IRPS)*, 2012. 139
- [123] P. Huang, B. Chen, Y. Wang, F. Zhang, L. Shen, B. Weng, Y. Tang, G.-Q. Lo, and D.-L. Kwong, “Analytic model of endurance degradation and its practical applications for operation scheme optimization in metal oxide based RRAM,” in *IEEE International Electron Devices Meeting (IEDM)*, 2013. 139
- [124] D. Veksler, G. Bersuker, L. Vandelli, A. Padovani, L. Larcher, A. Muraviev, B. Chakrabarti, E. Vogel, D. Gilmer, and P. Kirsch, “Random telegraph noise (RTN) in scaled RRAM devices,” in *IEEE International Reliability Physics Symposium (IRPS)*, 2013. 140, 141
- [125] K. Ito, T. Matsumoto, S. Nishizawa, H. Sunagawa, K. Kobayashi, and H. Onodera, “The impact of RTN on performance fluctuation in CMOS logic circuits,” in *IEEE International Reliability Physics Symposium (IRPS)*, 2011. 141
- [126] M. Zangeneh and A. Joshi, “Design and Optimization of Nonvolatile Multibit 1T1R Resistive RAM,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 8, pp. 1815–1828, July 2014. 142, 176
- [127] S. Shyh-Shyuan, C. Kuo-Hsing, C. Pang-Shiu, C. Yu-Sheng, F. Chen, S. Keng-Li, K. Ming-Jer, and T. Ming-Jinn, “Fast-Write Resistive RAM (RRAM) for Embedded Application,” *IEEE Design and Test of Computers*, vol. 28, no. 1, pp. 64–71, September 2010. 142, 176

## RELIABILITY-AWARE MEMORY DESIGN USING ADVANCED RECONFIGURATION MECHANISMS

---

- [128] A. Benoist, S. Blonkowski, S. Jeannot, H. Grampeix, J. Nodin, E. Jalaguier, L. Perniola, and B. Allard, “28nm advanced CMOS resistive RAM solution as embedded non-volatile memory,” in *IEEE International Reliability Physics Symposium (IRPS)*, 2014. 145, 153, 154
- [129] H. Yenpo, G. Huang, and L. Peng, “Dynamical Properties and Design Analysis for Nonvolatile Memristor Memories,” *IEEE Transactions on Circuits and Systems I*, vol. 58, no. 4, pp. 724–736, October 2010. 147
- [130] P. Pouyan, E. Amat, and A. Rubio, “Reliability Challenges in Design of Memristive Memories,” in *International Workshop on CMOS Variability (VARI)*, 2014. 149
- [131] P. Pouyan, E. Amat, and A. Rubio, “Insights to Memristive Memory Cell from a Reliability Perspective,” in *International Conference on Memristive Systems (MEMRISYS)*, 2015. 149, 197
- [132] C. Jeffery and R. Figueiredo, “Hierarchical fault tolerance for nanoscale memories,” *IEEE Transactions on Nanotechnology*, vol. 5, no. 4, pp. 407–411, July 2006. 160
- [133] M. Conroy, *A Collection of Dice Problems*, 2015. 164
- [134] P. Pouyan, E. Amat, and A. Rubio, “Statistical Lifetime Analysis of Memristive Crossbar Matrix,” in *International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS)*, 2015. 174
- [135] P. Pouyan, E. Amat, and A. Rubio, “Analysis and Design of an Adaptive Proactive Reconfiguration Approach for Memristive Crossbar Memories,” in *IEEE International Conference on Nanoscale Architectures (NANOARCH)*, 2015. 174
- [136] A. Mohammed, H. Aly, H. Fahmy, M. Mustafa, and K. Nabil, “Memristor-based memory: The sneak paths problem and solutions,” *Microelectronics Journal*, vol. 44, no. 2, pp. 176–183, February 2013. 176



## BIBLIOGRAPHY

---

- [137] C. Meng-Fan, S. Shyh-Shyuan, L. Ku-Feng, L. Chen-Hsin, F. Chen, S. Keng-Li, K. Tzu-Kun, K. Ming-Jer, and T. Ming-Jinn, “A High-Speed 7.2-ns Read-Write Random Access 4-Mb Embedded Resistive RAM (ReRAM) Macro Using Process-Variation-Tolerant Current-Mode Read Schemes,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 878–891, December 2012. 176
- [138] C. Ching-Yi, S. Hsiu-Chuan, W. Cheng-Wen, L. Chih-He, C. Pi-Feng, S. Shyh-Shyuan, and F. Chen, “RRAM Defect Modeling and Failure Analysis Based on March Test and a Novel Squeeze-Search Scheme,” *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 180–190, January 2014. 180
- [139] S. Hamdioui, M. Taouil, and N. Haron, “Testing Open Defects in Memristor-Based Memories,” *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 247–259, October 2013. 180
- [140] S. Kannan, J. Rajendran, R. Karri, and O. Sinanoglu, “Sneak-Path Testing of Crossbar-Based Nonvolatile Random Access Memories,” *IEEE Transactions on Nanotechnology*, vol. 12, no. 3, pp. 413–426, March 2013. 180
- [141] “Stanford RRAM model: online: <https://nano.stanford.edu/stanford-rram-model>.” xxiv, 194, 195
- [142] P. Pouyan, E. Amat, and A. Rubio, “Memristive Crossbar Design and Test in Non-adaptive Proactive Reconfiguring Scheme,” in *European Conference on Circuit Theory and Design (ECCTD)*, 2015. 197
- [143] Y. Xie, *Emerging Memory Technologies Design, Architecture, and Applications*. Springer, 2014. 202