UC Santa Cruz
UC Santa Cruz Electronic Theses and Dissertations

Title
Design and Analysis of Robust Variability-Aware SRAM to Predict Optimum Access-Time to Achieve Yield Enhancement in Future Nano-Scaled CMOS.

Permalink
https://escholarship.org/uc/item/9pv711jz

Author
Samandari-Rad, Jeren

Publication Date
2012

Peer reviewed|Thesis/dissertation

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

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

DESIGN AND ANALYSIS OF ROBUST VARIABILITY-AWARE SRAM TO PREDICT OPTIMAL ACCESS-TIME TO ACHIEVE YIELD ENHANCEMENT IN FUTURE NANO-SCALED CMOS

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

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Jeren Samandari-Rad

December 2012

The Dissertation of Jeren Samandari-Rad
is approved:

____

Professor Richard Hughey, Chair

____

Professor Sung Mo (Steve) Kang

____

Professor Jose Renau

____

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Copyright © by

Jeren Samandari-Rad

2012

Table of Contents

List of Figures vii

List of Tables xi

Abstract xii

Dedication xiv

Acknowledgments xv

I Introduction **1**

1 Motivations **2**

2 Literature Review **7**
- 2.1 Classical Models 8
- 2.2 More Advanced Models 9
- 2.3 Current/Recent Models 12
  - 2.3.1 Limitation on Parallel Slicing 23
  - 2.3.2 Limitation on Slice Width 23
  - 2.3.3 Limitation on the Operation Region 25

3 Contribution **28**

II SRAM Architecture, Operation, and Design Considerations **36**

4 Hierarchical Memory Architecture **37**
- 4.1 6T-cell Structure and Operation 38
- 4.2 6T-SRAM Array (one bank) Structure and Operation 39
- 4.3 6T-SRAM Array (Multiple Banks) Structure and Operation 41
- 4.4 Btline and Wordline Segmenting 43

5 SRAM Operation **47**
5.1 Read . . . 50
5.2 Write . . . 51
5.3 Access-time . . . 52
5.4 Hold . . . 53

III SRAM Design Considerations and Analysis **55**

6 Design Considerations and Analysis, Device **61**
6.1 D2D and WID variations . . . 61
6.2 Static Noise Margin (SNM) . . . 65
6.2.1 Hold Noise Margin . . . 73
6.2.2 Read Noise Margin . . . 74
6.2.3 Write Noise Margin . . . 74
6.3 Soft Error . . . 77
6.4 Negative Bias Temperature Instability (NBTI) . . . 77
6.4.1 Supply Voltage and Temperature Dependence . . . 89
6.4.2 Input Control in Static and Dynamic Operation . . . 91
6.4.3 Impact of NBTI on Process/Design) . . . 95
6.5 Hot-Carrier Injection (HCI) . . . 98
6.6 Single Electron Tunneling (SET) . . . 100

7 Design Considerations and Analysis, Power **102**
7.1 Impact of Temperature on Delay, Power, and Performance . . . 102
7.2 Temperature and Voltage Variation . . . 114
7.2.1 Supply Voltage Variation . . . 114
7.2.2 Temperature Variation . . . 115
7.2.3 PVT Variations and their Reduction Techniques . . . 119
7.3 IR-Drop, EM, and Ldi/dt . . . 135
7.4 Interconnect Challenges . . . 143
7.4.1 Overview of Interconnect . . . 144
7.4.2 Requirements of the interconnection materials . . . 146
7.4.3 Progress Trend and Future of Interconnect . . . 147
7.4.4 SPICE Model and Performance Metrics . . . 152
7.4.5 Existing and Future Interconnects . . . 156
7.4.6 Performance comparison between Cu/low-k, m-SWCNT Bundle, and Optical Interconnects . . . 166
7.4.7 Capacitively Driven Low-Swing Interconnect (CDLSI) . . . 172
7.4.8 Performance comparison between CDLSI, Cu/low-k, CNT, and Optical Interconnects . . . 174
7.5 Major Techniques for Leakage Control in Caches/SRAMs . . . 176
7.5.1 Lowering the Quiescent Vdd (Gated-Vss) . . . 177

7.5.2 Multiple Threshold CMOS (MTCMOS) 177
7.5.3 Drowsy Caches 178
7.6 Power, Leakage, and Energy Delay 178
7.6.1 Power Overview 178
7.6.2 Dynamic Power Consumption 179
7.6.3 Dissipation Due to Direct-Path Currents 184
7.6.4 Static Consumption 187
7.6.5 The Power-Delay Product, or Energy per Operation 192
7.6.6 Energy-Delay Product 193

IV Failure in SRAM **197**

8 Failure in SRAM **198**
8.1 SRAM cell failure 200
8.1.1 Read Failure 201
8.1.2 Write Failure 203
8.1.3 Access Failure 204
8.1.4 Hold Failure 205
8.2 Modeling Timing Errors 206
8.2.1 Our General Approach and Assumptions 207
8.2.2 Timing Errors in SRAM Memory 210

V Proposed Model: VAR-TX **212**

9 Our Proposed Model **213**
9.1 Derivation of access-time and its variation 217
9.1.1 D2D variability analysis 220
9.1.2 WID variability analysis 221
9.1.3 Combined WID and D2D analysis 231
9.2 Incorporating leakage, power, and area 232
9.3 Model assumptions and implementation 232
9.4 Model optimization 233
9.5 How to use the model 234

VI Experimental Results **235**

10 Simulation Results and Analysis **236**
10.1 Verification by Monte-Carlo 238
10.2 Validation of model optimization 241
10.3 Delay Simulation Results and Analysis 244
10.3.1 Access-time 244

10.3.2 Cumulative V_{th}, L, and V_{dd} Variability . . . 248
10.3.3 Individual V_{th}, L, & V_{dd} Variations . . . 252
10.3.4 Wordline vs. Bitline Variability . . . 255
10.3.5 Bank Variability . . . 257
10.3.6 FMAX Mean Variability . . . 261
10.3.7 Area vs. SRAM size . . . 263
10.3.8 Temperature Impact on Relative Switching Frequency . . . 264
10.4 Power Simulation Results and Analysis . . . 267
10.4.1 Overview . . . 267
10.4.2 Impact of Parameter Variations on Leakage Current . . . 268
10.4.3 Statistical Estimation and Distribution of Leakage Current in SRAM . . 272
10.4.4 Impact of Transistor Threshold Voltage (V_{th}) and Temperature (T) on Leakage Power . . . 274
10.4.5 Simulation Results for Power, Leakage, and Energy . . . 276
10.4.6 Probability Distribution of Total Power . . . 278
10.5 SRAM yield-estimation model . . . 281

VII Conclusion **283**

11 Summary **284**

12 Future Work **291**

Bibliography **296**

A Our Published Paper (in ISQED–2012) [147] **314**

List of Figures

2.1 Flow to divide a nonuniform gate into slices [193]. 15
2.2 Threshold variation under NRG and RNWE [193]. 15
2.3 6 Transistor SRAM Schametic with RC network [197]. 17
2.4 Different lithographic profiles from the same layout profile of SRAM with different depth of focus (DOF) [197]. 17
2.5 An example of filling missing measurements on wafer using the EM algorithm [145]. 18
2.6 Flow for generation of tolerance bands [15]. 20
2.7 Benefits of using tolerances with PWOPC [15]. 22
2.8 Linear and exponential dependence of I_{on} and I_{off} on V_{th} change, respectively [193]. 26

4.1 6 transistor (6T) storage cell. 38
4.2 SRAM Array-structured memory organization of one bank. 40
4.3 Hierarchical memory architecture. 42
4.4 Concept of Bitline Segmenting (Segmented Virtual Ground, SVGND). 44
4.5 Hierarchical word decoding architecture; Wordline Segmenting circuitry for one wordline. 46

5.1 6T *read* operation. 50
5.2 6T *write* operation. 51
5.3 6T *access* operation. 52
5.4 6T *hold* operation. 53
5.5 (Figure III-A) Classification of variations in IC Design. 57
5.6 (Figure III-B) 6 transistor (6T) storage cell (repeated for convenience). 59

6.1 Graphical method of characterizing *Static Noise Margin (SNM)* of an SRAM cell [5]. 67
6.2 Stable and *metastable* states of an SRAM cell with a DC noise offset applied to one side [5]. 70
6.3 Stable and *metastable* states of an SRAM cell with a DC noise offset applied to two sides [5]. 72

6.4 Comparison of *hold noise margin*, *read noise margin*, and *write noise margin* of 6T-SRAM designs [180]. 75
6.5 Variation of SNM and failure probability with (a) width of the access transistors; and (b) normalized cell area [115]. 76
6.6 An NBTI model [34] vs. measurement data by W. Wang et al. [182]. 82
6.7 Impact of V_{th} variation on NBTI. 83
6.8 NBTI timing analysis framework [184]. 85
6.9 Random input sequence. (a) Normal case. (b) Extreme case [184]. 86
6.10 Timing degradation analysis algorithm [184]. 88
6.11 Optimal V_{dd} for minimum degradation of circuit performance for two different 16-nm SRAM architectures: optimal ($\frac{64:64:16}{1:1:1}$) and non-optimal ($\frac{4:64:256}{1:1:1}$). 91
6.12 Delay degradation over time for various duty cycle sets of two sample circuits. . 94
6.13 Frequency degradation of an 11-stage ring oscillator (RO) under both process variation and NBTI effect [184]. 96
6.14 Example circuit to demonstrate the critical path changing with time. 97

7.1 6 transistor (6T) storage cell (repeated for convenience). 104
7.2 A piece of resistive material with electrical contacts on both ends [101]. 110
7.3 NMOS Mobility & Threshold, and wire Resistance change vs. Temperature. . . 111
7.4 Drain Current and Wire Delay vs. Temperature. 112
7.5 Supply voltage variation [27]. 115
7.6 Within die temperature variation [27]. 116
7.7 Optimal FBB for sub-90-nm generations [27]. 121
7.8 Leakage reduction by reverse body bias [27]. 122
7.9 Target frequency binning by adaptive body bias [27]. 123
7.10 Temperature based V_{cc}/frequency throttling [27]. 125
7.11 Measured delay changes to V_{cc} and Temperature [172]. 127
7.12 Impact of temperature on a commercial 65-nm technology [191]. 128
7.13 The 8T-SRAM cell architecture showing the WR and RD ports [143]. 131
7.14 Measured number of single bit failures in the 16 KB array with and without V_{cc} droop [143]. 133
7.15 IR-Drop & Tolerance vs. V_{dd} [62]. 139
7.16 Effectiveness of on-die decoupling capacitors [27]. 140
7.17 Electrical-thermal coupling. (a) Flow chart and (b) temperature-dependent resistivity of metals [155]. 142
7.18 Voltage Drop on Plane Shape [62]. 143
7.19 Schematic cross-section of backend structure, showing interconnects, contacts, and vias, separated by dielectric layers [148]. 145
7.20 Input Buffer Distribution [130]. 148
7.21 Delay as a function of technology node both for *global* interconnect and typical CMOS gate [87]. 150
7.22 Hillocks and voids induced by electromigration with high current density in a Cu interconnect [87]. 150

7.23 One segment of a distributed wire model using SPICE [87]. 152
7.24 Equivalent circuit of a distributed RC interconnect with step input function [87]. 153
7.25 Schematic illustration of the surface and grain boundary scatterings, and the barrier effect [87]. 157
7.26 Cu resistivity in terms of wire width taking into account the surface and grain boundary scattering and barrier effect [87]. 158
7.27 The impact of interconnect scaling [87]. 159
7.28 Three dimensional illustration of (a) SWCNT, (b) MWCNT [87]. 160
7.29 Transmission line LC components of SWCNT [87]. 160
7.30 (a) Inductance and resistance and (b) Inductance to resistance ratio as a function of the wire width [87]. 161
7.31 Graphical illustration of 2-D Graphene nano-ribbon (GNR) [56]. 162
7.32 Resistance comparison between GNR, mono-layer SWCNT, and Cu [2]. 163
7.33 (a) Schematic of an optimally buffered interconnect. (b) The equivalent circuit of one segment [87]. 164
7.34 Equivalent circuit model of a repeater segment for CNTs [87]. 165
7.35 The schematic of a quantum-well modulator-based optical interconnect [83]. . . 166
7.36 Latency as a function of technology node for two different interconnect lengths [125, 50]. 167
7.37 Energy per bit vs. technology node for two different interconnect lengths corresponding to *global* and *semiglobal* wire length scales [125, 50]. 168
7.38 Latency and energy per bit in terms of wire length for the 22-nm technology node [87]. 169
7.39 The impact of CNT and optics technology improvements on power density vs. bandwidth density [87]. 171
7.40 The impact of CNT and optics technology improvement on latency vs. bandwidth density [87]. 172
7.41 Schematic of conventional low-swing interconnect scheme [141]. 173
7.42 Conventional low-swing scheme with additional power supply [141]. 173
7.43 (a) Simple illustration of repeated capacitively driven low-swing interconnect (CDLSI). (b) Zoomed schematic of one segment of CDLSI. (c) Equivalent circuit model of one segment [87]. 174
7.44 Delay vs. bisectional bandwidth density (Φ_{BW}) [87]. 175
7.45 Energy Density vs. bisectional bandwidth density (Φ_{BW}) [87]. 176
7.46 Dynamic Dissipation due to Charging and Discharging Capacitances [141]. . . 180
7.47 Short-circuit currents during transients [141]. 185
7.48 Sources of leakage currents in CMOS inverter (for V_{in}=0 V) [141]. 188
7.49 Different components of SRAM cell leakage (based on Mukhopadhyay et al. [115]). 190
7.50 Normalized delay, energy, and energy-delay plots for CMOS inverter in 16-nm CMOS technology. 195

8.1 Read Failure: Flipping data during "*read*." 201
8.2 Write Failure: Memory cell does not register an input change correctly. 203

8.3 Access failure: $T_{ACCESS} > T_{LIMIT}$ 204
8.4 Hold failure: The destruction of the cell content in standby mode. 205
8.5 Example probability distributions. 208

9.1 Curve fitting for Hspice simulation for an SRAM. 223
9.2 Spatial correlation modeling for WID variations (Based on Fig.1 of Agarwal [4]).226

10.1 Spatial correlation modeling for WID variations (Based on Fig.1 of Agarwal [4]) (repeated for convenience). 239
10.2 Verifying our proposed model with Monte Carlo. 241
10.3 Validating optimization capability of our model. 242
10.4 Comparing the improved cumulative distribution function (CDF) of optimum-architecture Access-Time with its counterpart CDFs. 243
10.5 Access-time for "square" SRAM (*ACS*), Access-time for "non-square" SRAM (*ACI*), and *ACI* break-down traces. 246
10.6 Comparing the *ACI* (ideal access-time) 3-sigma corner points of 16-nm with those of 180-nm and 45-nm. 249
10.7 Cumulative distribution of access-time for 4 different SRAM sizes in 16-nm node.251
10.8 Individual Distribution of Access-time for (a) 180-nm 64KB SRAM and (b) 16-nm 64KB SRAM. 254
10.9 Wordline vs. Bitline 3σ corner-points (*ACH* and *ACL*) Variability of 16-nm SRAM. 255
10.10 Bank Variability; Access-time variation vs. number of banks. 258
10.11 Bank Variability; illustrating the distribution of *ACI* (ideal access-time) for two different organizations. 259
10.12 Area showing higher increase rate for each doubling of SRAM sizes, as compared to that of access-time. 263
10.13 Relative switching frequency versus temperature for different threshold voltages. 264
10.14 Probability distribution of the relative chip frequency as a function of V_{th}'s σ. . 266
10.15 Comparisons of the analytical model [195] against our circuit-level simulation results for 16-nm. 270
10.16 Distribution leakage of a 16-nm SRAM cell (I_{leak}). 273
10.17 Relative leakage power in the 16-nm SRAM chip as a function of V_{th}'s σ. . . . 275
10.18 Relative leakage power versus temperature for different threshold voltages at 125°C. 275
10.19 Read Dynamic Power, Standby Leakage Power, and Ideal Access-time (ACI) for different SRAM sizes in our 16-nm design. 276
10.20 Illustrating the combined "Read Dynamic Power + Standby Leakage Power" and the Total Read Dynamic Energy for different SRAM sizes in our 16-nm design. 277
10.21 Total Read Dynamic Energy and Ideal Access-time (ACI) for different SRAM sizes in our 16-nm design. 278
10.22 The probability distribution of the total power for four different SRAM sizes. . 280

List of Tables

6.1 Long term prediction Model of δV_{th} for both periodical and nonperiodical input sequence [184]. 86
6.2 Simulation results for two 16-nm SRAM circuits: *arcN* (non-optimum, $\frac{4:64:256}{1:1:1}$) and *arcO* (optimum, $\frac{64:64:16}{1:1:1}$) 89

7.1 Temperature dependency of mobility, threshold voltage and resistance [191]. . . 105
7.2 Temperature-induced delay change in a 65-nm technology [191]. 128
7.3 α and β for lumped and distributed networks for different points of interest [87]. 154

10.1 Comparison of different architectures with Ref. (VARIUS [169]). 244
10.2 Comparing the **cumulative** *ACI* 1-sigma of 16-nm with those of 180-nm and 45-nm for different SRAM-sizes. 250
10.3 Comparing the **individual** *ACI* 1-sigma of 16-nm with those of 180-nm and 45-nm for different SRAM-sizes. 253
10.4 Analysis of Mean and standard deviation of Ideal Access-Time (ACI) for two different organizations, in 16-nm SRAMs of different bank numbers. 260
10.5 FMAX (maximum frequency) MEAN Variability for a 64KB SRAM in three different technology nodes. 262
10.6 SRAM yield before and after optimization. 282

Abstract

DESIGN AND ANALYSIS OF ROBUST VARIABILITY-AWARE SRAM

TO PREDICT OPTIMAL ACCESS-TIME

TO ACHIEVE YIELD ENHANCEMENT

IN FUTURE NANO-SCALED CMOS

by

Jeren Samandari-Rad

Design variability due to inter-die (D2D) and intra-die (WID) process variations has the potential to significantly reduce the maximum operating frequency and the effective yield of high-performance chips in future process technology generations. This variability manifests itself by increasing the access-time variance and mean of fabricated chips.

This thesis proposes a new hybrid analytical-empirical model, called VAR-TX, that exhaustively computes and compares all feasible architectures subject to D2D and WID process variations (PV). Based on its computation, VAR-TX predicts the optimal architecture that provides minimum access-time and minimum access-time variation for yield enhancement in future 16-nm on-chip conventional six-transistor static random access memories (6T-SRAMs) of given input specifications and given area and power constraints. The given specifications include SRAM size and shape, number of columns, and word-size.

In addition, this thesis reviews 6T-cell design challenges and the main causes for failure. Also provided are several newly designed or modified circuits that are crucial for SRAM

stability, reliability, robustness, speed, and reduced power consumption. This thesis also compares the impact of D2D and WID variations on access-time for 16-nm SRAM with the 45-nm and 180-nm nodes and demonstrates that the drastic increase in the 1- and 3-sigma of the smaller nodes is mainly due to the increase in the WID variations. A considerable number of simulation results regarding access-time, leakage current, and dynamic power are presented and analyzed throughout this thesis to help predict the impact of process, operation, and temperature variations on SRAM variability, as well. Finally, the VAR-TX model argues previously published works that suggest that square SRAM always produces minimum delays and it significantly extends and enhances the older models by adding both an extra dimension of architectural consideration and additional device parameter fluctuation to the analysis, while producing delay estimates within 8% of Hspice results.

To my daughter Sonia, who has taught me how to love,

and to my adviser, Vice Provost & Professor Richard Hughey, who has dazzled me and so many others with not only his brilliance, but his amazing love and devotion towards all those around him;

I am forever grateful for the incredible impact you have made on my life.

Acknowledgments

I would like to thank all those who contributed to the emergence, creation, and correction of this thesis. I would like to start with the Lord for looking over my health and providing me with whatever I needed to complete my graduate education at UC Santa Cruz.

I would like to thank my thesis advisor, Professor and Vice Provost Richard Hughey, who is always an invaluable source of support and inspiration, from turning my research work around to giving me intelligent hints on effective research strategies, pointing me in the right direction, and providing me with a great deal of technical and editorial remarks/suggestions and so many answers I needed to complete my research project. I am immensely grateful to Prof. Hughey, whose intellectual, spiritual, and financial support made the success of this project possible. I will remain indebted to him for his vital support, his brilliance, and his unsurpassed positive attitude and personality for the years to come.

I would like to thank Professor Jose Renau for encouraging me to expand the design space of this project and to delve into several challenging award-wining related research works. His crucial suggestions helped me produce results that can be used by current and future SRAM designers. It‘s no wonder that many in and out of UCSC think of Prof. Renau as an embodiment of good heart and brain.

I would like to thank Chancellor and Professor Steve Kang for being kind enough to serve on my Thesis Defense committee and take time from his busy schedule (both at UC Merced and at UCSC) to read my thesis and give me his valuable feedback.

I am grateful to Prof. Matthew Guthaus whose initial ideas and direction helped me

get started on this project. Prof. Gauthaus' effective proofreading, his knowledge and expertise with conferences, and his patience in tolerating my numerous technical questions during the start of this project are among the reasons which helped my paper on SRAM (with him and Prof. Hughey) get the approval/acceptance of ISQED-2012 committee members.

I am grateful to Dr. Xuchu Hu (Cadence) for her smart solutions of the technical glitches I occasionally came upon, to Derek Chen (Space Systems/Loral) for his sound Ultrasim simulation tool hints, to Kevin Woo (Intel) and Ehsan Ardestani (UCSC) for answering my tricky LaTex questions, and to Dr. Rebekah Brandt (recent UCSC EE graduate) for her diligent proofreading contribution, which was instrumental in turning a rough draft into a user-friendly Thesis.

I apologize for the inadvertent potential omission of some deserving friends and colleagues whose contributions played a role in the extraordinary experiences I have been fortunate to enjoy. I thank them all, here, collectively.

Part I

Introduction

Chapter 1

Motivations

As device feature-size reduction is becoming dominant in the semiconductor industry, its impact on product reliability, yield, and therefore cost is dramatically increasing. Embedded microprocessors and other high-performance on-chip modules incorporate Static Random Access Memory (SRAM) or cache components that play significant roles in overall chip functionality and reliability. Unwanted variations in SRAM circuits may result in access-time variations and chip functional failures. This means the cost and performance of a vast number of chips today heavily depend on the reliability and speed of their on-chip SRAM, which is increasingly affected by scaled-down feature sizes.

The memory component of many chips span and even exceed 70% of the total area. Due to the crucial role of on-chip memories, much of the computer architecture research involves investigating trade-offs between various memory systems. This, however, can not be done adequately without a firm grasp of the costs of each alternative. For example, it is impossible to compare two different SRAM organizations without considering the difference in *access*

or *cycle-times*. Similarly, we must take the chip area and power requirements of each alternative into account. Only when all the costs are considered can we make an informed decision. But without a reliable, accurate, and inexpensive modeling tool in hand, this cost consideration itself would be either expensive, time consuming, inaccurate, or all three. This thesis provides an effective modeling methodology and corresponding toolkit that satisfies these requirements.

In order to continue the growth of modern memory technology, it is important to increase the access-time speed while curbing the energy usage. For faster access-time, new innovations in manufacturing processes and novel circuit designs are needed. Similarly, new efforts are required to control the power and energy consumption of storage, computing, and IT facilities and their cooling systems. Besides the environmental impact, excessive power consumption also reduces system reliability, increases cooling cost and cuts the battery cycle time. Effective power and thermal management will help to relieve the bottleneck of today‘s VLSI design and accelerate the growth of the information technology and many other similar industries. It will also enable today‘s computing and communication devices to work efficiently with emerging energy storage and energy harvesting technologies to achieve energy autonomy.

A robust, standard 6 transistor Static Random Access Memory (6T-SRAM) designed for an optimum architecture with power management considerations could significantly contribute to the system being able to work on different types of hardware with variable workload.

This thesis proposes a novel model (VAR-TX) that is suitable to the memory design of the next generation future technology node (i.e. 16-nm). It also covers recent progress on adaptive power management, including runtime monitoring, modeling, classification, learning, and controlling techniques for power and temperature optimization of a computing device. The

core of this thesis is presenting the process of building our proposed model (VAR-TX) that predicts the optimum architecture for a standard 6T-SRAM running at a maximum possible speed that satisfies a given power consumption and area for future technology nodes. However, to achieve this goal, it is necessary to cover several crucial stability-, reliability-, and energy-related topics that are considered (either explicitly or implicitly) during our SRAM design. This is because, like many other cutting-edge technologies, we believe that future technology nodes beyond 32-nm will face such challenges as temperature-related issues, the effect of Negative Bias Temperature Instability (NBTI), Hot Carrier Injection (HCI), the V_{dd} variation as a static IR drop or dynamic L di/dt, and several others (the most important of which are covered in this thesis) more than ever before. In a nutshell, our motivation for this research is to make the following contributions to the VLSI field:

★ Presenting VAR-TX: our new model that helps predict the variation of access-time due to process and operational variation in memory design for current and next generation future technology nodes (i.e., 16-nm).

★ Providing a first-order solution to mitigate the effects of increasing process variations in future technology nodes.

★ Providing an effective method to maximize the yield.

★ Making our proposed model VAR-TX freely available to the public to help predict the optimum architecture of a 6T-SRAM to achieve maximum speed for given power and area constraints.

★ Providing new simulation tricks that help avoid prohibitively long mixed-signal circuit simulations.

★ Providing a broad overview of the important challenges in SRAM design that could be used as a valuable reference for SRAM/cache designers.

These contributions are explained in further detail in Chapter 3. The following abstractivly lists our modeling methodology for the derivation of delay distribution, discussed in detail in Chapter 9.

1. Compute the sensitivities and store them in tables.
2. Compute the D2D component of the path delay.
3. Express the WID component of the path delay variation as an analytical expression of the device parameter variation.
4. Combine the two components (namely, D2D and WID) of the path delay variations to obtain the joint path delay distribution.
5. Optimize the delay through the examination of all possible architectures to achieve maximum yield.

The thesis is organized as follows:

In Part I, Chapter 2 begins by presenting literature research on prior approaches to memory compilers/models made for one or more of the following purposes: general trade-off analysis, analysis of tolerance to process variations, power reduction, and analysis of tolerance

of “soft errors” [transient errors induced by radiation] [17]. Part I, Chapter 3 states the contribution of this thesis to the SRAM community. Part II illustrates our hierarchical memory architecture (Chapter 4—in which several novel/modified circuits designed for increasing the speed, lowering the power, and minimizing the variability is presented and discussed). Part II also reviews SRAM memory operation (Chapter 5). Part III discusses design challenges. The design challenges and analysis is broken down into two separate chapters: Chapter 6 and 7. Chapter 6 covers such device-related topics as Die-to-Die (D2D) and within-in die (WID) variations, static noise margin (SNM), soft errors, negative bias temperature instability (NBTI), hot carrier injection (HCI), and single electron tunneling. Chapter 7 covers such power-related topics as temperature impacts, temperature and voltage variation, V_{dd} variation as a static IR drop or dynamic L di/dt, interconnect, techniques for leakage control, and the power (temperature, leakage, and energy-delay)—all of which contribute to the SRAM variability. The main causes for failure are discussed in Part IV (Chapter 8). Part V outlines the proposed new model VAR-TX (Chapter 9), after discussing two different classes of variability: inter-die (D2D) and intra-die (WID). Part VI, (Chapter 10) illustrates and analyzes our simulation results that demonstrate the impact of process (P), voltage (V), temperature (T), and technology nodes variability on speed, power, and yield of the designed SRAM. Part VII summarizes the impact of this research and future work. Finally, Appendix A presents this thesis’s published paper in ISQED–2012 [147].

Chapter 2

Literature Review

The scaling of SRAM in the presence of variability is becoming increasingly difficult, due to the reduced stability and increased leakage current with the scaling of silicon technology. Various circuit techniques have been proposed to curb process variations and thus improve SRAM access-time and stability while lowering power use. Past research on memory modeling can be classified into three groups, chronologically:

1. The *Classical Models* (oldest, circa 1990s) are primarily based on models and equations that take no variability considerations in mind.

2. The *more Advanced Models* (coming after the Classical Models) mostly focus on innovative ways to reduce delay, leakage/dynamic power, or a combination of these two.

3. Finally, the *Current/Recent models* (following the Advanced Models) are mostly based on the analysis of the effects of variability on the memory performance.

2.1 Classical Models

T. Wada et al. [167] present an equation for the access-time of an on-chip cache as a function of various cache parameters (cache size, **associativity***, block size) as well as organizational and process parameters. Unfortunately, Wada‘s access-time model has a number of significant shortcomings. First, the cache tag (a memory storage for holding addresses [131]) and comparator in set-associative memories are not modeled, and in practice, these often constitute the critical path. Second, each stage in this model (e.g., bitline, wordline) assumes that the inputs to the stage are step waveforms; actual waveforms are far from steps and this can greatly impact the delay of a stage. Third, all memory sub-arrays are stacked linearly in a single file; this can result in aspect ratios of greater than 10:1 and overly pessimistic *access-times*. Furthermore, Wada‘s decoder model is a gate-level model which contains no wiring parasitics. In addition, transistor sizes in this model are fixed independent of the load. As an example, the wordline driver is always the same size, independent of the number of cells that it drives. Finally, Wada‘s model predicts only the cache access-time, whereas both the *access-* and cycle-time are important for design comparisons.

---

* **Associativity** is a scheme used in memory architecture. Associativity allows each location in the main memory be cached by one of 2, 4, 8 or more cache locations. For example, in 2-way associativity, each location in the main memory could be in one of two cache locations. Associativity improves cache performance. For more see [131].

Among the proposals made in the recent past, CACTI [189] has been cited most. The CACTI authors improved Wada‘s access-time model [167] significantly by adding several new

features. These include a tag array model with comparator and multiplexer drivers. CACTI was an excellent analytical model for trade-off analysis in the late 1990s and early 2000s, but naturally exhibited shortcomings with scaled-down technology. Only the decoder component was modeled at the transistor level; remaining components were modeled at gate level or were equation-based. CACTI improved some of its shortcomings later on—in its newer versions (i.e. CACTI 6.5, 2009)—by modeling different types of wires, such as RC based wires with different power, delay, and area characteristics and differential low-swing buses. It also included, among others, a new feature of Non-Uniform Cache Access (NUCA) for chip multiprocessors that takes into account the effect of network contention during the design space exploration. Although much enhanced, as compared to its initial model, CACTI is still far from perfection. CACTI is based on DRAM technology and is mostly an equation-based model (and not hybrid empirical-analytical model like VAR-TX). It does not account for variations in V_{th}, L (also called L_{gate}), and V_{dd}, which greatly impact cache/SRAM stage delays and power; therefore, CACTI does not capture the effect of the random variations of electrical properties of the memory circuits on the access-time and power.

2.2 More Advanced Models

X. Liang and K. Turgay [98] present a unified architecture-level modeling methodology for SRAM and content-addressable-memory **(CAM*)** array structures. Although their model considers most fundamental circuit parameters, it cannot depict V_{th}, L_{gate}, and V_{dd} fluctuations over the entire SRAM.

* **Content-addressable memory (CAM)** is a type of computer memory used in certain high speed searching applications. It is also known as associative memory, associative storage, or associative array, although the last term is more often used for a programming data structure. Unlike standard computer memory (random access memory or RAM), in which the user supplies a memory address and the RAM returns the data word stored at that address, a CAM is designed such that the user supplies a data word and the CAM searches its entire memory to see if that data word is stored anywhere in it. If the data word is found, the CAM returns a list of one or more storage addresses where the word was found (and in some architectures, it also returns the data word, or other associated pieces of data). Thus, a CAM is the hardware embodiment of what in software terms would be called an associative array.

K. Agarwal and S. Nassif [6] offer an excellent model for characterizing the DC **noise margin*** of a memory cell; this model can estimate cell-failure probabilities during *read* and *write* operations. However, these authors do not show how parameter fluctuations, which are crucial to access-time, determine the stability of entire SRAMs of different sizes and shapes. The proposed VAR-TX model, driven by mixed-signal simulations of a standard 6T-SRAM circuit, does include these fluctuations.

---

* In electrical engineering, **noise margin** is the amount by which a signal exceeds the minimum amount for proper operation.

A. Agarwal et al. [4] present a useful model for path-based statistical timing analysis by modeling D2D and specially correlated WID device length variations. However, due to using the older 180-nm node, these authors neither included the impact of V_{th} and V_{dd} variations nor the architectural/organizational optimization in their modeling. This makes the application of their rather old model to the newer nodes (i.e. 32-nm and below) impractical and also makes their analysis and results much less accurate as compared to those of our proposed path-based model that takes all those missing factors into account.

R. Joshi et al. [70] propose a dynamic supply boosting technique for low voltage SRAMs at and beyond 65 nm using partially-depleted silicon-on-insulator **(PD-SOI*)** technologies. The technique exploits the capacitive coupling effect in a floating-body PD-SOI device to dynamically boost the virtual array supply voltage during *read* operation, thus improving the *read* performance, read/half-select stability, and V_{min} . Although their proposed technique enables significant reduction of the standby cell power and circuit active power in a single supply methodology, it requires a more complex circuitry and a special manufacturing process. It is also possible to improve V_{min} by using dual supply methodologies as discussed in [70, 71], but this comes at the expense of extra supply and wire routing complexity, both at the global and local levels.

*** Partially-depleted silicon-on-insulator (PD-SOI)** refers to a Semiconductor CMOS (complementary metal-oxide-semiconductor) process with seven layers of copper (Cu) interconnect and low-k dielectric.

M. Yamaoka et al. [103] propose either expanding the *write* margin, using a power-line-floating *write* technique, or process-variation-adaptive *write* replica circuit to enable low-voltage *write* operation. Although effective in considerably lowering the leakage power, these techniques require careful and sensitive control of both column select and row select to prevent the degradation of stability of other cells in the same row or column.

B. Mohammad et al. [111] use a novel circuit to increase the Static Noise Margin (SNM) and the *write* margin of the SRAM cell. Despite their success in increasing the SNM and in reducing the voltage swing of the circuit mostly during the *write* (but not necessarily during the *read* operation as well), the paper reveals that the speed of their memory *access* is

reduced in part due to their "W1 voltage reduction."

G. Ming et al. [110] suggest reducing the power consumption by dynamically charging the bitlines, as well as charge sharing due to bitline charge/discharge; but this comes at the expense of reduced static noise margin.

2.3 Current/Recent Models

Several good works regarding process variability have been published by P. Gupta in the recent past. In his earlier publication [60], Gupta proposes reducing the leakage power (and leakage power variability) by about 24%–38% by applying gate-length biasing only to those devices that do not appear in critical paths. This comes at the cost of up to a 10% delay penalty, thus assuring negligible degradation in the system level chip design performance. In his successor work [61], Gupta proposes algorithms for the creation of isolated and dense variants for each library cell to compensate for reduced delay and increased leakage incurred by lithography focus problems to achieve designs that are more robust to lithography focus variation.

Gupta complements his previous works with a new proposal [97] that suggests a new method to exploit the unequal drive and leakage current distributions across the transistor channel in order to find an optimal non-rectangular shape for the channel to achieve further savings in leakage current. More specifically, Gupta et al. propose making a library of two different cells: one for improved delay (with a shorter dumbbell-shape transistor channel, during I_{on}), and the other for improved leakage (with a longer dumbbell-shaped transistor channel, during I_{off}). Following that, in response to any last minute developments of the chip manufacturing

process that could cause specification failures, Gupta et al. present a new framework to perform an Engineering Change Order (ECO) to correct the problems through incremental gate sizing for process changes late in the design cycle.

In one of his latest works, Gupta et al. [34] address the main NBTI-induced degradation issues. They argue that the recent related works [34] that have relied on device-level analytical models are limited in their flexibility to model the impact of architecture-level techniques on NBTI degradation. He and his co-authors propose a flexible numerical model for NBTI degradation that can be adapted to better estimate the impact of architecture-level techniques on NBTI degradation. In this work, Gupta et al. shows that **guardbanding*** may still be an efficient way to deal with aging. Although insightful, especially for technology nodes prior to 45-nm, Gupta‘s work mostly hinges upon the systematic variation of gate-length (and gate-width) and not on the significance of random variation of V_{th} as well. Since the random variation of V_{th} is the dominant variability factor in newer technology nodes (i.e. 45-nm and beyond), the application of Gupta‘s analytical works (assuming V_{th} as constant) to the newer nodes may fall short of high accuracy and effectiveness.

---

* Traditionally, **guardbanding** has been used to protect against NBTI. For example, the operating frequency is reduced or supply voltage is increased to account for degradation over the lifetime of a design, such that there are no timing violations due to aging during the lifetime. The subject of NBTI is discussed in Chapter 6.

Mukhopadhyay et al. [115] offers an excellent model for failure probabilities of SRAM cells due to process-parameter variations. However, their computationally-intensive model only considers random fluctuations in V_{th}, and only for a single SRAM cell. Furthermore, they sug-

gest that their model could be improved by including systematic fluctuations in V_{th}, as well as considering both types of fluctuations (random and systematic) in L_{gate}.

Teodorescu et al. [169] build upon Mukhopadhyay‘s work [115] by modeling a selected group of 6T-cells in an array of 6T-cells, but still only include variation in V_{th}. Our VAR-TX model, in contrast, not only includes variations in V_{th}, L_{gate} and V_{dd}, but does so for an entire 6T-SRAM.

Among the contemporary reputable variability-related research works in academia are those developed by Yu Cao and his research group at Arizona State University. They create the Predictive Technology transistor Models (PTM) that this thesis has used for simulation. In one of their recent works [193], Y. Cao et al. develop an *efficient SPICE simulation method* and *statistical variation model* that accurately predicts threshold variation as a function of dopant fluctuations and gate length change caused by lithography and the etching process. By understanding the physical principles of atomistic simulations, they: 1) identify the appropriate method to divide a nonuniform gate into slices, as shown in Figure 2.1, in order to map those fluctuations into the device model; 2) extract the variation of V_{th} from the strong-inversion region instead of the leakage current, benefiting from the linearity of the saturation current with respect to V_{th}; 3) propose a compact model of V_{th} variation that is scalable with gate size and the amount of dopant and gate length fluctuations; and 4) investigate the interaction with non-rectangular gate (NRG) and reverse narrow width effect **(RNWE*)**.

---

* **RNWE** (reverse narrow width effect) nonuniformly reduces the threshold voltage in different locations: the closer a gate slice is to the gate end, the larger the drop is. Such nonuniformity along the width direction interacts with NRG and varies the output current [157, 159]. For instance, when the slice with the minimum length is close to the gate end extension (Shape 1 in Figure 2.2),

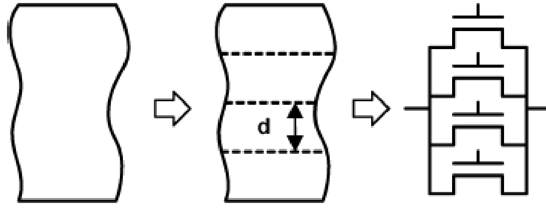


Figure 2.1: Flow to divide a nonuniform gate into slices. Each slice has a unique $V_{\mathrm{th}\,i}$ and L_i due o RDF and LER [193].

the threshold drop in that slice will be more significant due to both drain induced barrier lowering (DIBL) and stronger RNWE, leading to the largest leakage increase; on the other hand, if the slice with the minimum length is located far away from the gate end extension (e.g., in the middle of the gate, see Shape 2 in Figure 2.2), then RNWE is much weaker and the leakage is lower. Figure 2.2 shows these two representative conditions of the gate shape distortion, in which both shapes have the same nominal size and magnitude of NRG and line edge roughness (LER); but one is convex and the other is concave and thus, they are different in RNWE.

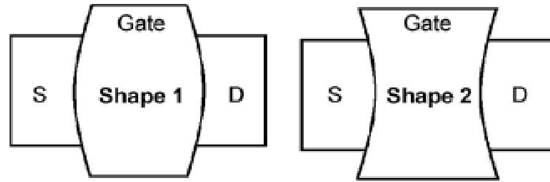


Figure 2.2: Threshold variation under NRG and RNWE. Two representative gate distortions under NRG [193].

To model a nonrectangular gate in the SPICE environment, the slicing method splits the nonuniform edge into many slices, such that each slice can be approximated into a regular transistor with a uniform gate length. One can then apply the nominal device model to each slice for predicting the I-V characteristics. The final performance of the transistor under LER is calculated from the summation of currents from all the slices [159, 59, 164]. This procedure is illustrated in Figure 2.1.

This proposed work [193] correctly models the variation of device output current in

all operating regions (given the post-lithography gate geometry) and projects the amount of V_{th} variation at advanced technology nodes. Although this method is rudimentary, easy to operate in practice, and widely adopted in previous works [193, 159, 59], it comes with some limitations: limitation on parallel slicing, limitation on slice width, and limitation on the operation region. Due to their conceptual usefulness, these three topics are briefly discussed in further detail at the end of this chapter (Subsections 2.3.1 – 2.3.3). In these three sections we will see how the three limitations can make the proposed modeling and method somewhat costly and prone to inaccuracy, if sufficient care is not taken.

The most respected industrial works on variation are from the IBM Austin Research Labs group, many of which authored or co-authored by Sani Nassif. The remainder of this section lists several of these works.

In one of the recent works from the IBM Labs group, Y. Zhou et al. [197] perform a critical study of the effects of Back-end-of-line (BEOL) lithographic variations on 45-nm SRAM performance and yield analysis. They present an SRAM simulation model with internal cell interconnect RC parasitics (see Figure 2.3) for their study of the BEOL lithographic impact. Using their method, they systematically evaluate the impact of BEOL variations on memory designs. First, they study the impact of ideal parasitics assuming no lithographic variations. Then they look into the worst-case, best-case, and nominal lithographic variations (see Figure 2.4) to show that on average, ideal parasitics impact the delay by more than 20-30% and also impact the stability yield leading to an increase of 100 mV to the SRAM minimum operating voltage, V_{min} . Based on these results, they claim that power estimation with their BEOL model is more accurate, and a traditional model without interconnect parasitics may be off by 33% in accuracy.

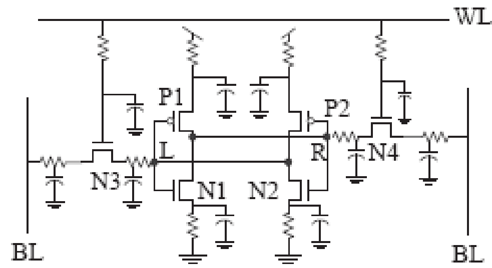


Figure 2.3: 6 Transistor SRAM Schametic with RC network [197].

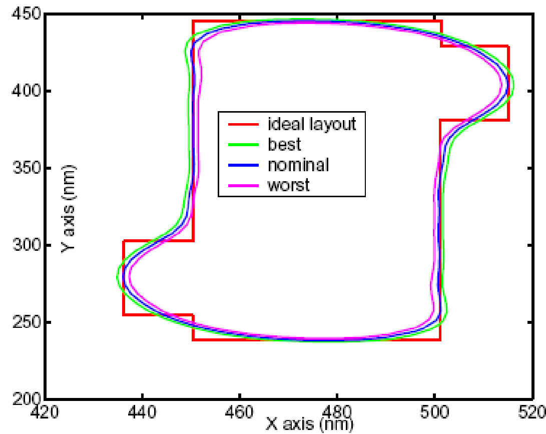


Figure 2.4: Different lithographic profiles from the same layout profile of SRAM with different depth of focus (DOF) [197].

The close match between these findings and the simulation results of our model (VAR-TX) further validates the analysis presented in this thesis. Y. Zhou et al. also show that the additional accounting of the lithographic variations for the BEOL study induces about 4% variation on the SRAM *read* delay. Finally, they point out that when the resistance change (due to misalignment) is of the same order of magnitude as the nonlinear device resistance, the impact is more severe.

Another recent work from the IBM Labs group [145], developed by Sherief Reda and Sani R. Nassif, proposes a novel statistical framework to model the impact of process

variations on semiconductor circuits through the use of process sensitive test structures. Based on multivariate statistical assumptions, they propose the use of the expectation-maximization algorithm (commonly known as EM) to estimate any missing test measurements and to calculate accurately the statistical parameters of the underlying multivariate distribution.

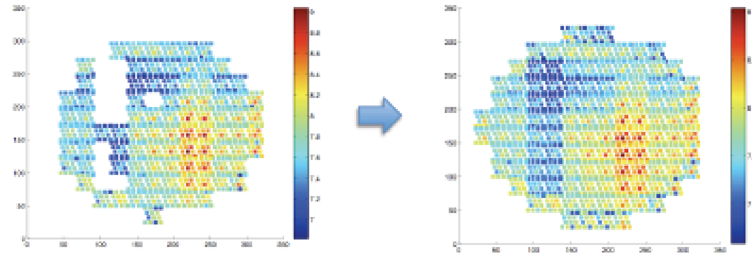


Figure 2.5: An example of filling missing measurements on wafer using the EM algorithm [145].

Figure 2.5 shows an example where the EM algorithm fills the missing measurements of one of the wafers. The color of a measurement gives its value (or speed in this case). Visual inspection shows that predicted values seem to "fit" within the range of the rest of the measurements. Using their proposed model, they analyze the impact of the systematic and random sources of process variations to reveal their spatial structures. They utilize the proposed model to develop a novel application that significantly reduces the volume, time, and costs of the parametric test measurements procedure without compromising its accuracy. They verify their models and results on measurements collected from more than 300 wafers and over 25,000 die fabricated at a state-of-the-art facility and prove the accuracy of their proposed statistical model and demonstrate its applicability towards reducing the volume and time of parametric test measurements by a factor of about 2.5 - 6.1 at no impact to test quality.

In another IBM work, they reason that the analysis performed at the "schematic" level

can be deceiving (as it ignores the interdependence between the implementation layout and the resulting electrical performance). In response, A. Bansal et al. [16] present a computational framework, referred to as "Virtual SRAM Fab," for analyzing and estimating pre-Si SRAM array manufacturing yield considering both lithographic and electrical variations. They demonstrate their proposed framework for SRAM design/optimization for the 45-nm node and use it for both the 32-nm and 22-nm technology nodes, as well. The authors illustrate the application and merit of the framework using two different SRAM cells in a 45-nm PD-SOI technology, which have been designed for similar stability and performance, but exhibit different parametric yields due to layout and lithographic variations. They also demonstrate the application of Virtual SRAM Fab for prediction of layout-induced imbalance in an 8T-cell, which is a popular alternative candidate for SRAM implementation in 32- and 22-nm technology nodes.

A few of the works from the IBM Labs group aim to attack the variability issues by proposing new lithography-related methodologies. As the move to low-k1 lithography has made it increasingly difficult to print feature sizes which are a small fraction of the wavelength of light, many of the manufacturing processes still treat a target layout as a fixed requirement for lithography. However, in reality layout features may vary within certain bounds without violating design constraints. The knowledge of such tolerances, coupled with models for process variability, can help improve the manufacturability of layout features while still meeting design requirements. Noticing such a notion, S. Banerjee et al. [15] propose a methodology to convert *electrical slack* in a design to *shape slack* or tolerances on individual layout shapes using a two-phase approach. In the first step, the *delay slack* is redistributed to generate *delay bounds* on individual cells using linear programming. In the second phase, which is solved

as a quadratic program, these *delay bounds* are converted to *shape tolerances* to maximize the process window of each shape. The authors show that the *shape tolerances* produced by their proposed methodology can be used within a process-window optical proximity correction (PWOPC) flow to reduce delay errors arising from variations in the lithographic process.

The authors validate the accuracy of their proposed methodology by presenting the results of their experiments on 45-nm SOI cells using accurate process models that show that the use of their *shape slack* generation in conjunction with PWOPC reduces delay errors by a factor of 2 on average (i.e. from 3.6% to 1.4%), compared to the simplistic way of tolerance band generation. Figure 2.6 illustrates the two key components in the depicted flow of the proposed methodology.

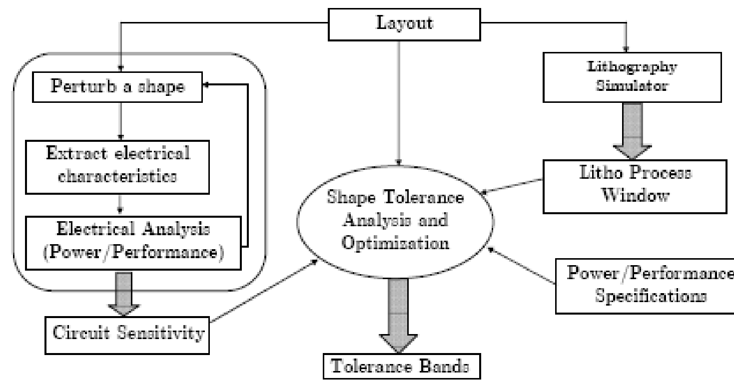


Figure 2.6: Flow for generation of tolerance bands [15].

One of the key components is *Electrical sensitivity* and the other one is the *lithographic process window*. *Electrical sensitivity* is a measure of how critical a particular shape is from the design point of view. Some examples of critical shapes are transistors and interconnects on timing-critical paths. Variations in manufacturing that perturb the electrical properties of these shapes may have an adverse effect on the timing of the design. In order to improve para-

metric yield, the tolerances on such shapes is required to be small. Conversely, the *lithographic process window* is a measure of the degree of difficulty in printing a certain shape [102]. The smaller the process window for a shape, the more difficult it is to print in the presence of process variability. Some examples of shapes with low *lithographic process window* are line-ends and layout hot-spots [86]. Such shape constructs require greater flexibility (higher tolerances) in order for lithography to find a robust solution.

Figure 2.7 shows a transistor with a small outer tolerance and a large inner tolerance. This condition is typical of devices on critical paths. By this figure, the authors in IBM group [15] intend to show that they have performed both **OPC*** (optical proximity correction) and **PWOPC*** on this feature. They also show that they have subsequently generated lithographic contours at different process corners and compiled the process variability (PV) band which represents the outermost and innermost aerial image contours in the presence of variability. Finally, and most importantly, the authors want to show that whereas the use of OPC cannot ensure that contours across the process window will lie within acceptable *shape tolerances*, the use of PWOPC moves the PV bands to lie within the *shape slack*; thus validating their proposed methodology.

*** Optical proximity correction (OPC)** is the technique of generating a mask to print a given layout [43]. A conventional OPC tool typically uses optical and resist models to predict the image of the mask on the wafer. The tool then computes the edge placement error (EPE) between the image and target and finally moves mask edges so as to minimize this geometric error. This technique optimizes the image at a single (nominal) point and hence does not provide a solution that is robust to variations in the lithographic process.

*** Process-window OPC (PWOPC)** is a mask generation technique that increases lithographic yield by improving image quality at multiple process corners [15]. This method computes the aerial image contours at a number of different lithographic process points

and uses a weighted sum of EPE as the cost function for minimization. When tolerances are specified, the algorithm optimizes for weighted EPE until a contour at a certain corner exceeds the bounds, at which point the computational effort shifts to optimization at that corner alone [57].

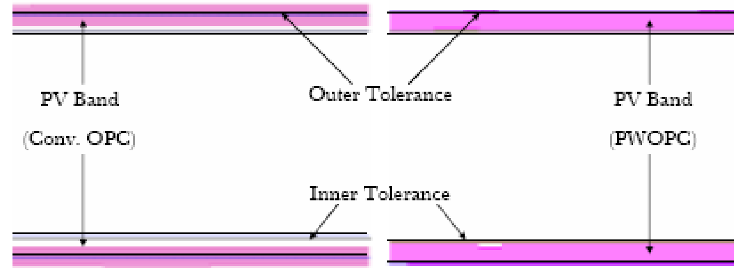


Figure 2.7: Benefits of using tolerances with PWOPC [15].

Finally, to extend the performance-based SRAM application space of a nominal 1 V technology, from the traditional higher voltage high-speed domain [47, 135, 185], to the half-volt domain for low-power computing, handheld, and mobile applications—in addition to addressing the tightened energy budget for server class memories—the IBM labs group has recently released another paper [90]. In this paper, J. Kuang et al. report a high-performance, dual *read* port, 8-way set associative 6T-SRAM, with a one clock cycle *access* latency, in a 32 nm metal-gate PD- SOI process technology, for low-voltage applications. Dual *read* port 6T-SRAMs play a critical role in high-performance cache designs; thanks to doubling of *access* bandwidth even though it comes at the cost of some stability and sensing challenges which typically limit the low-voltage operation. The authors propose a hardware that exhibits a robust operation at 348 MHz and 0.5 V with a *read* and *write* power of 3.33 and 1.97 mW, respectively, per 4.5 KB active array when both *read* ports are accessed at the highest switching activity data pattern. The authors show that the hardware is also capable of producing an *access* speed of 1.2

GHz, but at a slightly higher voltage of 0.6 V.

2.3.1 Limitation on Parallel Slicing

This is the first of the three *Limitations of the Gate Slicing Method* (mentioned in Section 2.3). By partitioning the nonuniform gate into parallel slices along the source-to-drain direction (see Figure 2.1), the first underlying assumption is that the current in each slice maintains the same direction from source to drain, i.e., there is no significant distortion of the electrical field along the channel direction. Otherwise, there would be a pronounced amount of current across the slice boundary and the slicing method is not able to provide a correct prediction under LER [136, 159].

With the aggressive down-scaling of both channel length and channel width, more physical effects, such as DIBL and the fringe field from the gate edge, will affect the channel region. The distortion of the electric field may be exacerbated in the extreme case. If the current along the width direction becomes comparable to the current along channel direction, then the gate slicing method has to be corrected.

2.3.2 Limitation on Slice Width

This is the second of the three *Limitations of the Gate Slicing Method*. Even if the assumption of parallel slicing is true, there are still fundamental limitations on slice width in this approach [193]—especially when the effect of random dopant fluctuations (which usually requires atomistic simulation to provide sufficient accuracy) is considered. We can classify the limitation on slice width as *Upper Bound of Slice Width* and *Lower Bound of Slice Width*,

described below.

Upper Bound of Slice Width: The spatial frequency of LER

There are many factors that cause LER during the sub-wavelength lithography and the etching process. These different factors lead to different spatial frequencies and amplitudes of the distortion of the gate edge. Using the silicon data of gate length change under LER [44], Cao et al. [193] show two regions of LER with distinct spatial frequencies: the high-frequency region (HF) that has a **characteristic length*** smaller than 5 nm and a low frequency region (LF) that has a characteristic length larger than 10 nm [44]. The exact values of their characteristic lengths depend on the fabrication technology. When we split a nonuniform gate under LER, the width of each slice needs to be smaller than the characteristic length in order to track the change in gate length with adequate accuracy. For instance, to model a typical LER gate, the slice width should be smaller than 20 nm. This phenomenon defines the upper bound of gate slice width during the slicing.

***Characteristic length**, if not defined, refers to the autocorrelation length, which is defined as the length at which the autocorrelation function of the random channel potential decays by a factor of e^{-1} [11].

Lower Bound of Slice Width: Random dopant fluctuations

Due to the random position of dopants in the channel, V_{th} exhibits an increasing amount of variation with the continuous scaling of transistor size [11]. For a relatively long channel device, this behavior is well recorded in Pelgrom‘s model [134]. However, as the channel length is approaching the length scale of the fluctuation, such atom-level

randomness can no longer be represented by a V_{th} model in the subthreshold region—which is the statistical average of the potential in the channel. Such an average is not able to track the atomistic change [11, 134]. In order to apply the slicing approach to a compact V_{th}-based device model, the slice width must be larger than the correlation length of random channel potential near the threshold. This length is typically around several nanometers, depending on the doping concentration [11]. Only when both the upper and lower bounds of the slice width are satisfied, the partition of a single LER transistor is meaningful in predicting the current in all regions. Within this limitation, the slicing method is only valid in the case that the correlation length of LER is larger than the correlation length of random potential due to RDF (random dopant fluctuation). Upon the emergence of new advances in the etching process leading to the reduction of the LER correlation length, the method to track LER shape should be revised.

2.3.3 Limitation on the Operation Region

This is the third of the three *Limitations of the Gate Slicing Method*. After appropriately slicing the gate with a non-rectangular shape, the characteristic of each slice can be described using compact device model. The summation of all the slices provides the behavior of the original LER gate. For the nominal condition, each slice has a different V_{th} from the deterministic effects of narrow-width and DIBL, which lead to the increase in the leakage current and the reduction in the effective gate length. The changes of I_{on} and I_{off} under these effects are sufficiently captured through the equivalent gate length (EGL) model [159], i.e., a smaller Lmin for I_{off} and a larger Lmax for I_{on}. In their work, Cao et al. [193] follow the same modeling

approach to formulate the nominal transistor model. However, the situation becomes more complicated when they incorporate statistical variation due to random dopant fluctuation into each slice. Since I_{off} is an exponential function of V_{th} (see Figure 2.8), which is very nonlinear, the linear superposition of I_{off} from each slice is not applicable and thus, the mean and distribution of V_{th} cannot be extracted from the statistical analysis in the subthreshold region [193]:

$$mean\ \ of\ \ exp\left(-\frac{V_{\text{th}}}{nkT/q}\right) \neq exp\left(-\frac{mean\ \ of\ \ V_{\text{th}}}{nkT/q}\right) \tag{2.1}$$

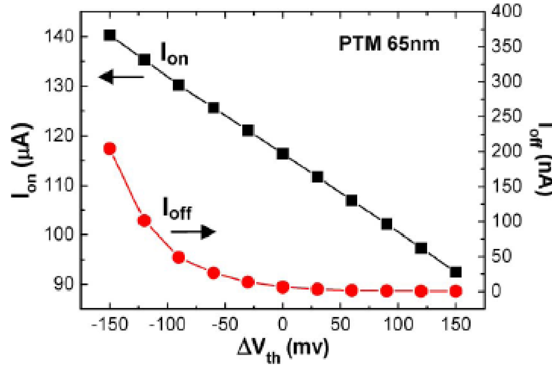


Figure 2.8: Linear and exponential dependence of I_{on} and I_{off} on V_{th} change, respectively [193].

To overcome this barrier and still maintain the mathematical correctness, the linearity of I_{on} has to be leveraged to study the statistics of V_{th}. For a short-channel device, I_{on} has a linear dependence on V_{th}, due to strong velocity saturation [196]. This behavior is illustrated in Figure 2.8 for PTM 65-nm technology. The linearity of I_{on} is even stronger in scaled CMOS devices [196]. As a result, the limitation that fails the statistical V_{th} extraction from I_{off} (see Equation (2.1)) is removed. The strong linearity of I_{on} provides a well-behaved basis to study V_{th} variation under RDF in all cases of LER, and therefore allows using an I_{on}-based method to ex-

tract V_{th} variation, embed it into the nominal device model, and then predict I_{off} change [193]. However, we should note that the inaccuracy of an I_{off}-based extraction method also depends on the size of the transistor: as the slice becomes smaller, the V_{th} variation increases; therefore, the error caused by the nonlinearity (see Equation (2.1)) is more pronounced. On the other hand, if the slice size is large enough, then the differences among slices become smaller and the I_{off}-based modeling error is reduced. For complete analysis of limitatations on slice width the reader is encouraged to consult Cao et al. work [193].

Chapter 3

Contribution

This chapter presents the contributions of this thesis research to the SRAM modeling community. Since prior works—several of which were introduced in the previous chapter (Literature Review)—neither incorporated the role of the SRAM architecture in the optimization of 6T-SRAM performance prediction nor considered the important impact of the process and environment variations (threshold voltage, transistor length, supply voltage and temperature) concurrently a need for such model is both necessary and providential.

- Prior models, like CACTI [189], are typically based on an abstract or courser-grained gate or equations models, while failing to incorporate the critical impact of the manufacturing process variations on the memory performance. The application of these older models to today‘s circuits, which exhibit a high degree of fluctuations in their electrical characteristics, is no longer practical. Therefore, we propose a new model that extends previous models and fixes many of their shortcomings. Our proposed model for 6T-SRAM circuits is completely at the transistor level, with all transistors being subject to manufacturing

process variations. Our model also includes layout parasitics (e.g., the resistance and capacitance of all the bitlines (wires) and wordlines (wires) in the 6T-cell array). A model built at such a highly detailed level is, unsurprisingly, capable of mimicking the behavior of today‘s SRAMs. This is one of our reasons for doing this research.

- Prior methods and models either solely rely on one SRAM cell (e.g., Mukhopadhyay [115], Nassif [197]), on a few cells (e.g., VARIUS [169], Nassif [16]), or simply use ADDER or FO4 (fan-out four) in their modeling of SRAM components (e.g., VARIUS [169]). None of these methodologies can illustrate the variability distribution of speed, power, and performance of 6T-SRAMs as accurately as the model which considers the critical path of all the cells in 6T-SRAM arrays with their components actually designed rather than simply modeled by ADDER or FO4. This explains our second reason for presenting this thesis.

- Prior methods and models focus on only one or two of the parameters causing variability. For example Gupta et al. [60] focus only on L_{gate} variations assuming a constant threshold. Similarly, Nassif et al. [193] investigate the impact of lithography imperfections on threshold variations without including the impact of other variability factors such as supply voltage and temperature in their simulation results. These models and methods, therefore, can not fully capture the electrical fluctuation impact of all the process and environment parameter variations on the performance of 6T-SRAMs. This justifies our third reason for undertaking this research: Our model takes into account all the above factors plus the additional architectural aspect of SRAMs to achieve a more realistic analysis of SRAMs variability.

- ♦ Prior works did not consider all possible 6T-SRAM architectures subject to NBTI, HCI, temperature, supply voltage, threshold voltage, and transistor length variations in their variability analysis. Therefore they cannot match the accuracy of our suggested VAR-TX model as regards SRAM performance and yield. This constitutes our fourth reason for this research.

Design variability due to D2D and WID process variations has the potential to significantly reduce the maximum operating frequency and the effective yield of high-performance chips in current and especially in future process technology generations. This variability manifests itself by increasing the leakage and access-time variance and mean of fabricated chips.

In two recent models [192, 169], path-based variation-induced statistical timing analyses of SRAM memories were proposed. Although insightful, neither of these or other subsequent approaches capture the architectural dependence of the gate delay due to variability of fan-out gates; nor do they address the WID and D2D variability of V_{dd} (which we confirm is not as significant as threshold and transistor length). The former case, in particular, is important in selecting the architecture that reduces both the delay and the delay variation and hence increases the yield while meeting given area and power constraints.

In this thesis, therefore, we propose VAR-TX: a new path-based approach to statistical timing analysis that considers both the architecture- and process-variations. We model variations of the gate delay due to fluctuations of the input slope and output loads resulting from variations of fan-in and fan-out stages in the path for all possible 6T-SRAM architectures. We propose a model where the D2D and architecture-dependent WID variations of all the major

parameters of the device are modeled as two separate components. Furthermore, we propose efficient methods for computing path delay variability due to either source, as well as their combined effect.

Specifically, this thesis makes the following major contributions, shown below under two separate headings, namely, "Thesis Contributions in Brief" and "Thesis Contribution in Detail," for a quick glimpse and a detailed review, respectively.

Thesis Contributions in Brief

★ We propose a novel hybrid analytical-empirical model VAR-TX that helps predict the minimum delay and/or minimum delay variation in current and next generation on-chip memories.

★ Our VAR-TX model provides a first-order solution to mitigate the effects of increasing process variations in future technology nodes, while providing results that are within 8% of Hspice.

★ Our VAR-TX model helps predict the optimum architecture that helps maximize the yield.

★ Our model VAR-TX contradicts previously published works that suggest square SRAM always give minimum delays.

★ Additionally, we present the access-time and power variations calculated by our model for the future 16-nm node and compare it to those of the recent 45-nm and older 180-nm nodes.

- ★ By publishing this thesis, we are making our proposed modeling methodology freely available to the public. As a bonus, we are also making the associated toolkit/software of our proposed model VAR-TX freely available to the public upon request (through email request; jeffsrad@soe.ucsc.edu). The VAR-TX toolkit predicts the optimum architecture of a 6T-SRAM to achieve maximum speed for a given power and area constraint.

- ★ The proposed model and analysis method that was applied to standard 6T-SRAM in this thesis provides the ground work for its extension to other types of memory such as 8T-, 10T-, or multi-ported SRAM, cache and CAM in a straightforward manner for future work.

- ★ This thesis gives a broad overview of the important challenges in SRAM design and could be a valuable reference for SRAM designers.

- ★ By sharing our model and analytical method for free with the VLSI design community, we are providing a fast and accurate method for long mixed-signal circuit simulations, which will hopefully increase the success of future circuit designs.

Thesis Contributions in Detail

- ■ We propose a novel hybrid analytical-empirical model VAR-TX that exhaustively computes and compares the sensitivity of different 6T-SRAM architectures to the variations in threshold voltage (V_{th}), gate length (L), and supply voltage (V_{dd}). This enables the user to select the optimal architecture that gives the minimum delay and/or minimum delay variation while providing the maximum yield possible, for the given area and power

constraints. In considering the sensitivity of the critical path to variations in both the overall architecture and within the individual devices, we not only add a new dimension to the path-based statistical timing analysis but also significantly improve upon the previous *access-times* models [4, 192, 115, 169]—which neither considered architectural sensitivity nor all three parameter variations. The proposed model yields delay and power estimates within 8% of Hspice results for the circuits we have designed.

- Using our model, we argue previously published works that suggest square SRAM always produce minimum delays. We show that minimum access-time and/or access-time variation can be obtained from a non-square SRAM.

- Additionally, we present the access-time and power variations calculated by our model for the future 16-nm node and compare it to those of the recent 45-nm and older 180-nm nodes. We also present several other experimental and simulation results to show the larger impact of process variations in increasingly small devices and therefore help shed light on the challenges of future robust circuit design.

- By publishing this thesis, we make the theory behind our model freely available to the public to provide the memory designers of today and the next generation with an accurate modeling methodology that can be useful for first-order trade-off analysis in the early stages of memory design. Additionally, and as a bonus, we make the associated software of our proposed model VAR-TX freely available to the public upon request (through sending email request to the author: jeffsrad@sbcglobal.net). This provides the memory designers of today with an accurate toolkit that can help ease the difficult and expensive

task of selecting the optimum organizations for given specifications and help predict the associated range of variations of access-time, all in the early stages of design. For example, an SRAM/cache designer or computer architect can use our proposed model to readily estimate the delay or the power and area cost for pushing an SRAM of a given specification to its maximum speed. These specifications include the combination of such user-entries as SRAM size (in bits), SRAM shape, the number of columns, and required bandwidth (number of SRAM outputs in bit).

- We hope that our proposed hybrid analytical-empirical methodology will inspire VLSI circuit designers and researchers to resort to new and innovative simulation methods and tools similar or even more advanced than those we have used to avoid the prohibitively long simulation times that result when numerous critical parameters are varied throughout large circuits. One such tool is Ultrasim (from Cadence Inc.) and another one that is becoming more popular is SOlidus—which is a tool for managing the impact of variations on design. SOlidus is typically used in conjunction with TSMC (an analog mixed-signal PDK tool that provides an alternative solution to the existing traditional design flow) and Virtuoso (a design and test EDA tool from Cadence) to improve the yield and centering (tighter distribution) results with fewer Monte Carlo samples and shorter simulation time for the same level of coverage.

- The proposed model and analysis method that was applied to standard 6T-SRAM in this thesis provides the ground work for its extension to other types of memory such as 8T-, 10T-, or multi-ported SRAM, cache and CAM in a straightforward manner for future

work.

- This thesis gives a broad overview of the important challenges in SRAM design and could be a valuable reference for SRAM designers.

Part II

SRAM Architecture, Operation, and Design Considerations

Chapter 4

Hierarchical Memory Architecture

SRAM Overview

Static random access memory (SRAM) is a type of semiconductor memory. The word static indicates that, unlike dynamic RAM (DRAM), SRAM does not need to be periodically refreshed, as SRAM uses bi-stable latching circuitry to store each bit. SRAM exhibits data reminiscence, but is still volatile since data is eventually lost when the memory is not powered. A typical SRAM is composed of several blocks, called banks. Each bank has an array of memory cells and also several periphery devices of its own that help access the memory cells in the array. Each memory cell (bit-cell) stores one bit of data. For successful low voltage SRAM operation, various bit-cell topologies with 5 transistors (5T-cell), 6 transistors (6T-cell), 8 transistors (8T-cell), or 10 transistors (10T-cell) have been proposed [91, 13]. Considering the overall performance and design density, 6T-SRAM is the conventional choice for most on-chip memory designs.

Figures 4.1 to 4.5 illustrate the overall organization of a conventional 6T-SRAM. Go-

ing from bottom to top, the schematic for the 6T-cell, the overall organization of a conventional 6T-SRAM array of one-bank, and then of multiple-banks, are shown and discussed in the next three sections of this chapter. The block diagram of our bitline- and wordline-segmenting are illustrated and discussed in the subsequent sections of this chapter.

4.1 6T-cell Structure and Operation

The six-transistor static random access memory cell (6T-SRAM) is the conventional choice for most on-chip memory designs. With power applied, SRAM provides permanent data storage. Figure 4.1 shows the schematic for the 6T cell of a 6T-SRAM.

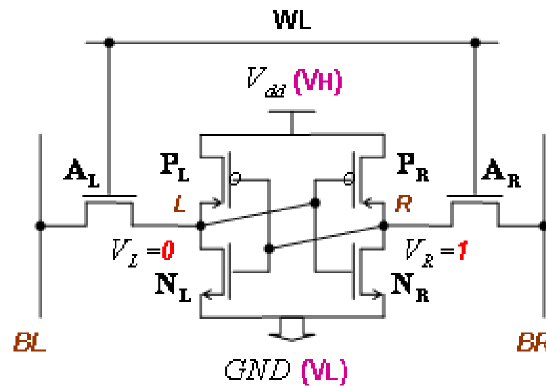


Figure 4.1: 6 transistor (6T) storage cell.

As shown in Figure 4.1, each bit in an SRAM cell is stored on four transistors ($N_L - P_L$ and $N_R - P_R$) that form two cross-coupled inverters. This storage cell has two stable states which are used to denote 0 and 1. Two additional access transistors (A_L and A_R) serve to control the access to the storage cell during *read* and *write* operations. The wordline (WL in Figure 4.1) controls the two access transistors—which, in turn, control whether the cell

should be connected to the bitlines, B_L and B_R. This pair of bitlines is used to transfer data for both the *read* and *write* operations. Although it is not strictly necessary to have two bitlines, both the signal and its inverse are typically provided in order to improve noise margins and access time.

Cell design requires a complex balance among several factors including speed, silicon area, and power/leakage consumption [111, 9, 60]. The balancing task is challenging due to conflicting interactions among several factors, which will be explained in further detail in Chapter 6.

4.2 6T-SRAM Array (one bank) Structure and Operation

Figure 4.2 illustrates the overall organization of a conventional 6T-SRAM composed of only one bank. The main *components* include the row and column decoders, the precharge, wordline and bitline segmenting circuitry, the 6T-cell array, the sense amplifier, *write* circuitry, output drivers, an internal-clock, and pre-decoders for wordline and bitline segmenting.

The array contains as many bitline pairs as there are columns multiplied by word-width. The array also contains as many wordlines as there are rows in the array. Within the active bank array, only one set of bitline pairs and only one segment of the selected wordline can go high at a time. Each memory cell along the selected row segment is associated with a pair of bitlines; each bitline is initially precharged high. Our sense-amplifiers are shared among several pairs of bitlines through a set of multiplexers inserted before the sense amps; the select lines of the multiplexer are driven by the column decoder. The number of bitlines

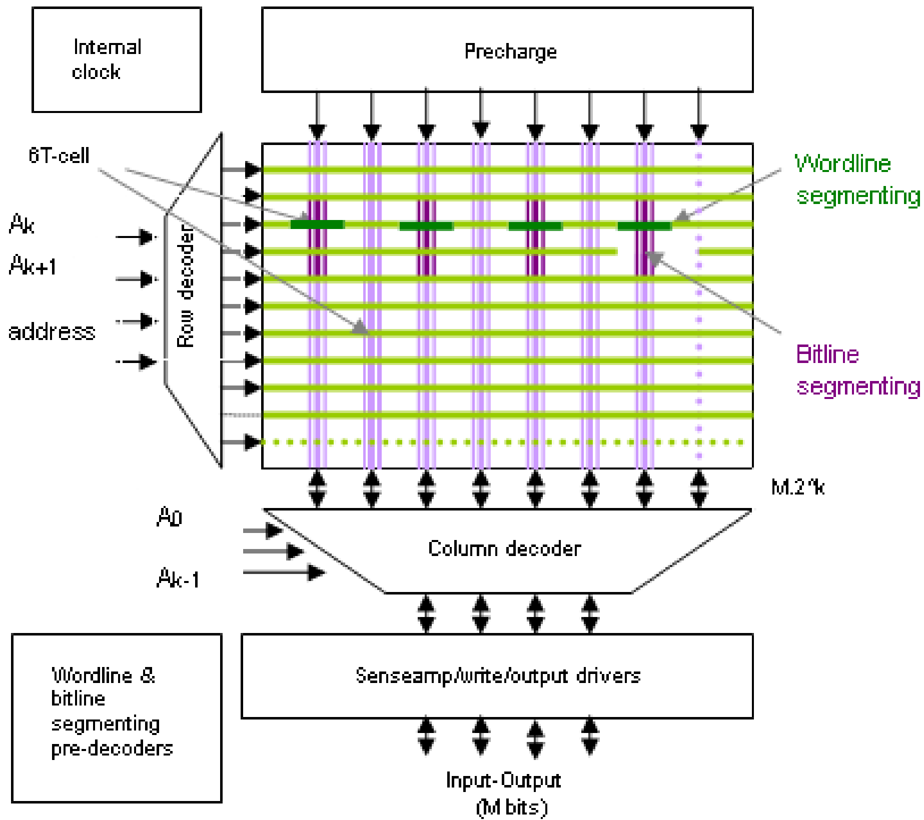


Figure 4.2: SRAM Array-structured memory organization of one bank.

that share a sense-amplifier depends on the number of columns in the array. The information read from the sense amplifiers is sent to the output by the output drivers. The parameterized internal-clock circuitry (not shown) determines the width, speed, slew-rate, and the intervals in between the pulses that activate the precharge, word lines, and sense amplifier circuitries. The column-decoder and the row-decoder decode the memory address.

The row is selected by pulling up one of the wordline-segmenting rows and the column is selected by precharging the bitlines associated with that column. Subsequently, the bitline-segmenting component pulls the source voltage of the 6T-cell pull-down transistors from

VL (in the range of 0.2 – 0.4V) to GND in the selected segments located along the selected set of bitline pairs. Next, the wordline-segmenting component chooses one segment of the selected wordline in the array by driving it high. The schematic and block-diagrams of bitline- and wordline-segmenting are shown in Section 4.4.

As shown in Figure 4.1, when a wordline goes high, each memory cell in the selected segment of that row pulls down one of its two bitlines; the value stored in the memory cell determines which bitline goes low. Each sense-amplifier monitors a pair of bitlines and detects when one changes. By detecting which line goes low, the sense-amplifier can determine the contents of the selected memory cell, and pass that information to the output driver.

In our design, we modified bitline segmenting [156] by replacing the large NMOS transistor with a smaller pass transistor pair for the virtual ground switch (vgs) of the Column Virtual Ground (CVG). The vgs of the CVG connects the CVG to ground when it is activated. We also modified wordline-segmenting [181] by using variable-size logical effort buffers in place of constant size combinational logic. Moreover, we modified Rabaey‘s [141] techniques for bank organization by using a bank-decoder to feed combinational logic in place of muxs. These modifications contribute to maximizing the speed, minimizing the static and dynamic power consumption, or lowering the circuit parameter fluctuations.

4.3 6T-SRAM Array (Multiple Banks) Structure and Operation

The architecture of Figure 4.2 works well for memories in a range of 64–256 Kbits. Larger memories start to suffer from a serious speed degradation as the length, capacitance,

and resistance of the wordline and bitline become excessively large. Larger memories have consequently gone one step further and added one extra dimension to the address space, as illustrated in Figure 4.3.

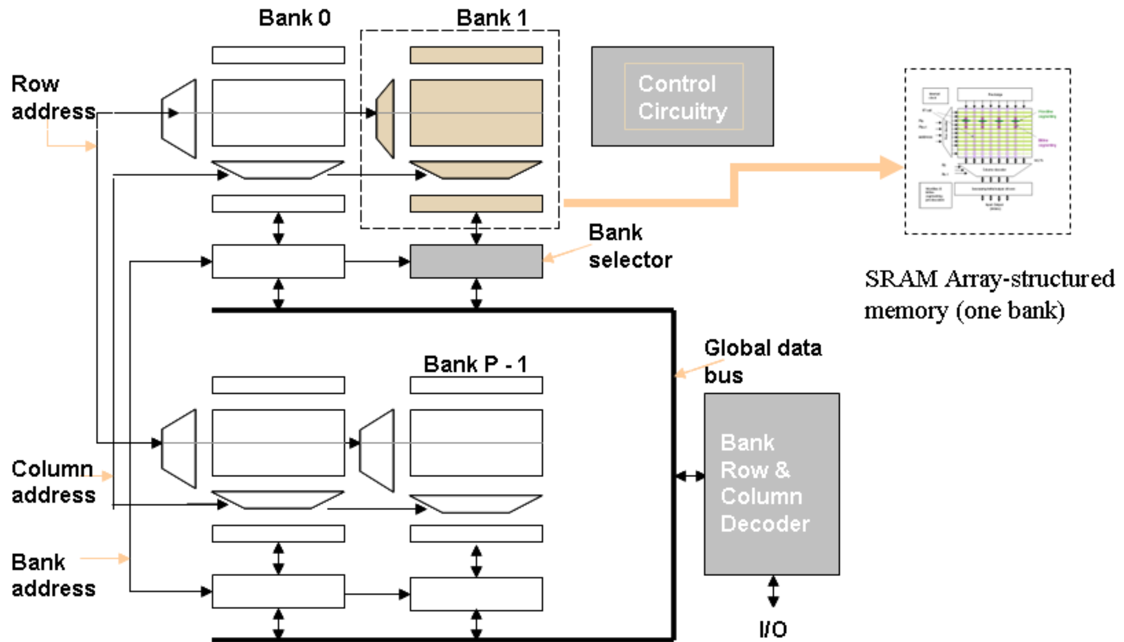


Figure 4.3: Hierarchical memory architecture. The bank selector enables a single memory bank at a time.

Figure 4.3 shows the hierarchical memory architecture of an SRAM composed of several banks. The bank selector enables a single memory bank at a time while all other banks remain in sleep mode as part of power variability management.

The memory is partitioned into *P* smaller banks. The composition of each of the individual banks is identical to that of Figure 4.2. A word is selected on the basis of the row and column address that are broadcast to all of the banks. An extra address word, called the bank address, selects one of the *P* banks to be read or written. This approach has a dual advantage:

1. The length of the local wordlines and bitlines within the banks are kept within bounds, resulting in faster access times.

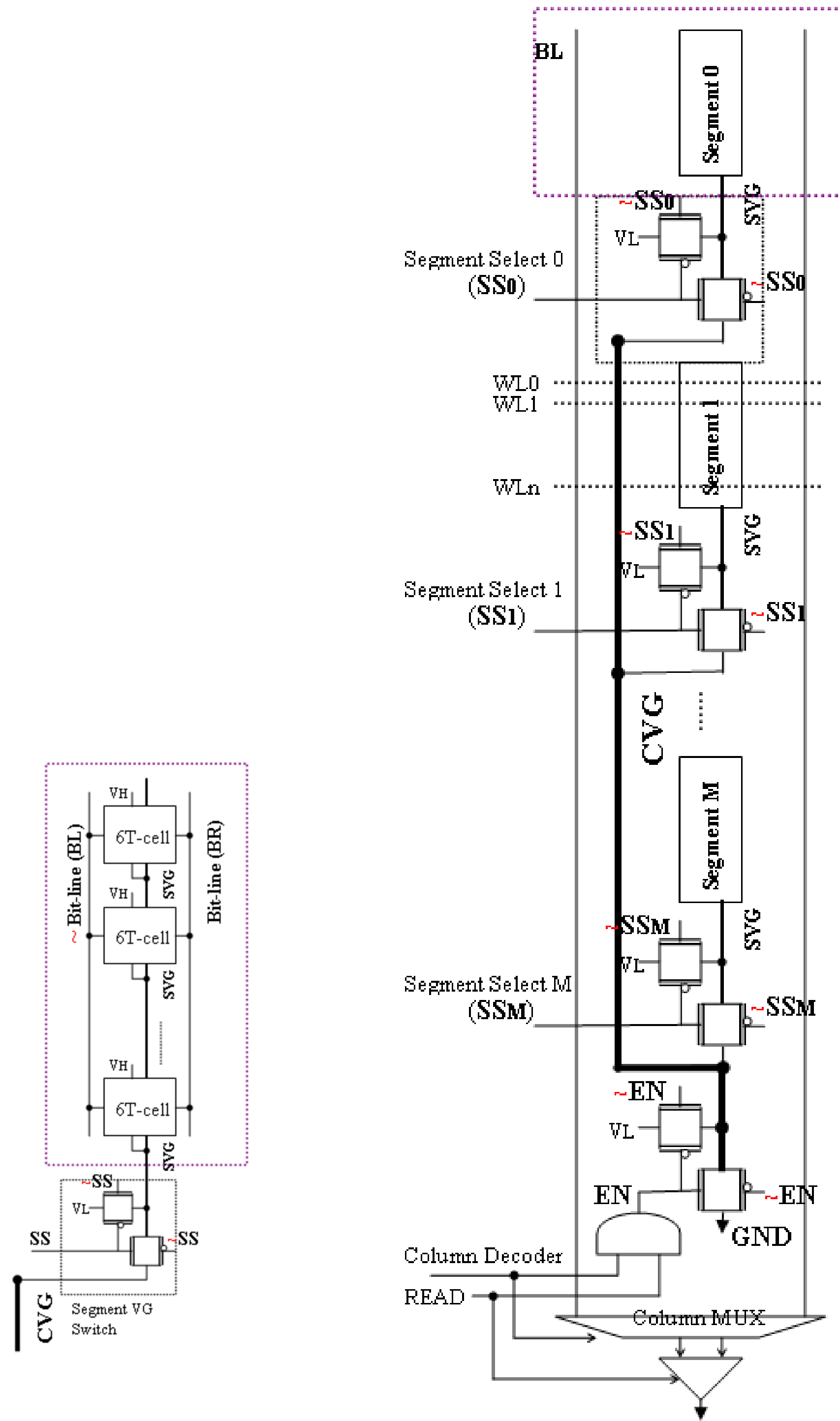
2. The bank address can be used to activate only the addressed bank. Non-active banks are put in the power-saving mode with pre-charge, row and column decoders, sense amplifiers, and other peripheral devices disabled. This results in substantial power savings, which is a major concern in very large memories [141].

4.4 Btline and Wordline Segmenting

This section discusses the concept of bitline- and wordline-segmenting. The architecture of the SRAM array is based on segmentation of the memory cells in a column and in a row, as shown in Figure 4.4 and 4.5, respectively. Both bitline-segmenting and wordline-segmenting schemes play a major role in decreasing the access time and power consumption of SRAM.

To avoid fully pre-charging the un-selected bitlines (to save power), our design uses a pre-column decoder. Similarly, to reduce the access/cycle time and static/dynamic power consumption, our design uses a wordline-segmenting organization. This allows the activation of only the selected cells on the selected row, rather than the activation of all the cells on the selected row. Additionally, our bitline segmenting organization allows the activation of only the small number of the selected segments/cells of the selected bitlines, while putting the rest of the array (a large number of unselected segments/cells) into sleep mode.

Figure 4.4(a) shows one of the several segments in a column, and Figure 4.4(b) shows the entire bitline-segmenting scheme for one pair of bitlines.



(a) Architecture of one Bitline Segment (b) Architecture of several Bitline Segments

Figure 4.4: Concept of Bitline Segmenting (Segmented Virtual Ground, SVGND).

There are as many bitline-segmenting circuitries as there are bitline pairs in each memory array. A segment in a column is defined as a set of cells on the same column with a shared segment virtual ground (SVG). A virtual ground switch connects the virtual ground of the segment to the virtual ground of the column (CVG). The CVG is a node shared between all virtual ground switches on the same column (see Figure 4.4(b)).

If the virtual ground switch of a segment is activated by its SS signal going high (see Figure 4.4(b)), the virtual ground voltage of the segment is equal to the virtual ground voltage of the column; otherwise, the virtual ground of the segment keeps its nominal voltage VL (Voltage Low, which is larger than ground voltage and is about 0.2V to 0.4V). The logic of the virtual ground switch is an inverter which drives the SVG node either to VL or to the voltage of the CVG node depending on the control signal (which is either VL or almost equal to ground voltage "0V", respectively). This bitline-segmenting architecture, which is a modification of that introduced by M. Sharifkhani [156], allows a significant reduction in both delay time and power consumption.

Figure 4.5 shows the hierarchical wordline segmenting architecture. The architecture is composed of pre-signal boosting, divided-wordline (DWL) structure, and the post-signal boosting (logical effort). The word select line is divided into multiple segments. The number of hierarchies in the pre- and post-signal boost circuitry is determined by the total load capacitance of the word decoding path. In this example, the global wordline distributes the pre-boosted signals to the control logics (AND) of all 6T-cells located on the same row and assigned to the same column number. Similarly, yet more restrictedly, the local wordline distributes the post-signal boost to WL of only a selected number of 6T-cells located on the same row and designated to

the same column number. This wordline segmenting architecture, which is our modification of that introduced by P. Wang [181], realizes a significant reduction in both delay time and power consumption.

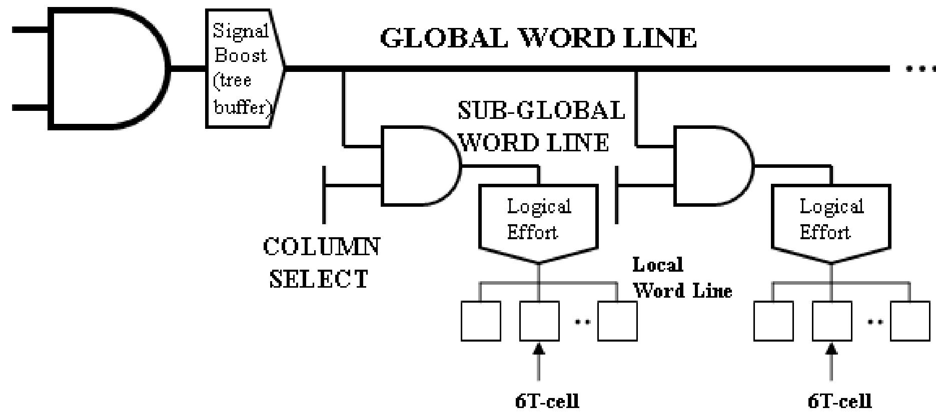


Figure 4.5: Hierarchical word decoding architecture; Wordline Segmenting circuitry for one wordline.

Chapter 5

SRAM Operation

An SRAM 6T-cell has four different states it can be in: *Read*—when the data has been requested; *Write*—when updating the contents; *Access*—when one of the bitlines is discharging during *read*; *Hold* (standby mode)—when the circuit is idle. The SRAM in *read* mode or *write* mode should have "readability" and "write stability," respectively. This means the content of the 6T-cell must not be flipped during *read* (i.e.: changing from 1 to 0 or from 0 to 1), and must not be updated by the opposite of the intended bit during *write* (i.e.: overwritten by 1 when instructed to be overwritten by a 0 or vice versa). Similarly, the SRAM in *access* mode or *hold* mode should have "speed capability" and "hold retain ability," respectively. This means discharging of the bitline up to a certain level (i.e.: $V_{\rm dd}/2$) must not take longer than an allowable time (T_{limit}) during *access* and the content of the 6T-cell must not be changed during *hold* (due to the smaller voltage across the cell for leakage reduction purposes).

For our first-order analysis of *read*, *write*, and *access* operation of our 6T-cell, we

used the following unified MOS model for manual analysis [141]:

$$
\begin{aligned}
& I_D = 0 \quad for \quad V_{GS} - V_{\text{th}} \leq 0 \\
& I_D = \mu C_{ox} \frac{W}{L} [(V_{GS} - V_{\text{th}}) V_{min} - \frac{V_{min}^2}{2}](1 + \lambda V_{DS}) \quad for \quad V_{GS} - V_{\text{th}} \geqslant 0 \\
with \quad & V_{min} = min(V_{GS} - V_{\text{th}}, V_{DS}, V_{DSAT}), \\
and \quad & V_{\text{th}} = V_{th0} + \gamma(\sqrt{|-2\phi_F + V_{SB}|} - \sqrt{|-2\phi_F|})
\end{aligned}
\tag{5.1}
$$

During the *read*, *write*, and *access* operations, the current I_D flows through two of the 6 transistors in a 6T-cell, typically with one transistor in saturation and the other one in triode mode. Equations (5.2) and (5.3), which are derived from Equation (5.1) show the drain current I_D for a transistor in saturation and triode mode, respectively.

Triode mode or linear region (also known as the ohmic mode)

When $V_{GS} > V_{\text{th}}$ and $V_{DS} < (V_{GS} - V_{\text{th}})$

The transistor is turned on, and a channel has been created which allows current to flow between the drain and the source. The MOSFET operates like a resistor, controlled by the gate voltage relative to both the source and drain voltages. The current from drain to source is modeled as:

$$I_D = \mu_n C_{ox} \frac{W}{L} [(V_{GS} - V_{\text{th}}) V_{DS} - \frac{V_{DS}^2}{2})] \tag{5.2}$$

where μ_n is the charge-carrier effective mobility, C_{ox} is the gate oxide capacitance per unit area, W is the gate width, and L is the gate length. The transition from the exponential

subthreshold region to the triode region is not as sharp as the equations suggest.

Saturation or active mode

When $V_{GS} > V_{\textbf{th}}$ and $V_{DS} > (V_{GS} - V_{\textbf{th}})$

The switch is turned on and a channel has been created, which allows current to flow between the drain and source. Since the drain voltage is higher than the gate voltage, conduction is not through a narrow channel but through a broader, two- or three-dimensional current distribution extending away from the interface and deeper into the substrate. The onset of this region is also known as **pinch-off** due to the lack of channel region near the drain. The drain current is now weakly dependent upon drain voltage and controlled primarily by the gate-source voltage. It can be modeled approximately as:

$$I_D = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}[(V_{GS} - V_{\text{th}})^2(1 + \lambda(V_{DS} - V_{DSAT})) \tag{5.3}$$

The additional factor involving λ, the channel-length modulation parameter, models the current dependence on drain voltage due to the *Early Effect*, or channel length modulation.

Both *read* and *write* operations start with decoding/translating the input memory address to a 1-hot code. The four different states work as follows:

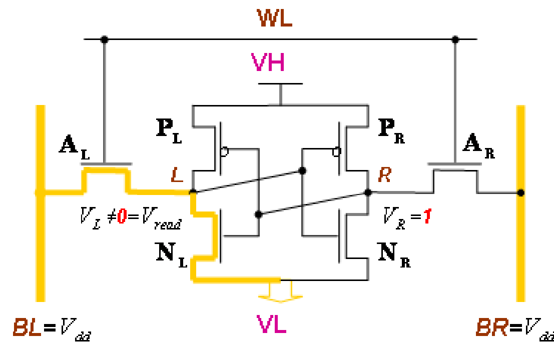


Figure 5.1: 6T *read* operation.

5.1 Read

Looking at Figure 5.1, assume that the content of the memory is a **1**, stored at R. With the input address already decoded, the *read* cycle is started by precharging both the bitlines to a logical **1**, then WL is asserted which enables both the access transistors. The second step occurs when the values stored in R ($V_R = 1$) and L ($V_L = 0$) are transferred to the bitlines by leaving BR at its precharged value of **1** and discharging BL through A_L and N_L to a logical **0**. On the BR side, the transistors P_R and A_R pull the bitline toward V_{dd}, a logical **1**. If the content of the memory were a **0**, the opposite would happen and BL would be pulled toward **1** and BR toward **0**. Then BL and BR, with a small potential difference of delta between them, reach a sense amplifier, which performs differential signaling to distinguish which bitline has a higher voltage and thus indicates whether there was a **1** or $\neq$**1** stored. The sensitivity of the sense amplifier determines the speeds of the *read* operator (more sensitive = faster). The sense amplifier, subsequently, passes this information to the output driver.

The role of bitline segmenting and wordline segmenting during the *read* is better understood by looking back at Figure 4.4 and 4.5, respectively. In the *read* mode, the CVG

signal is pulled down before the selected wordline segment and SS signals are asserted. Then, after the selected wordline segment is set high while the SVG is kept low, the selected SRAM cell discharges the appropriate bitline. During this interval, the voltage across the cell is VH (Voltage High, about 20% less than V_{dd}), therefore, the cell is capable of discharging the bitline. A selective discharge of the SVG node to ground of only one segment in the array during the *read* operation prevents the discharge of both internal capacitances of the neighboring cells on the same row and the internal capacitances of the non-accessed segments on the same column. Therefore, it saves a significant amount of power.

5.2 Write

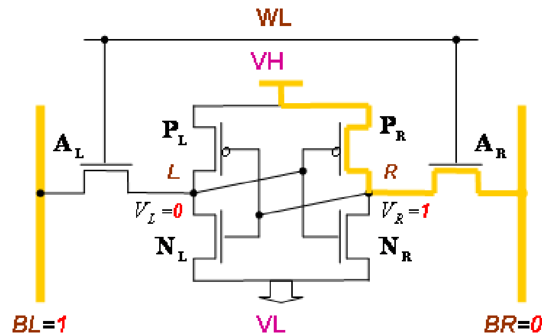


Figure 5.2: 6T *write* operation.

Looking at Figure 5.2, assume that the content of the memory is a **1**, stored at R. With the input address already decoded, the *write* cycle begins by applying the value to be written to the bitlines. If we wish to write a **0**, we would apply a **0** to BR and a **1** to BL. This is similar to applying a reset pulse to an SR-latch, which causes the flip-flop to change state. A **1** is written by inverting the values of the bitlines. WL is then asserted and the desired value is stored.

The reason this works is that the bitline input-drivers (*write* circuitry) are designed to be much stronger than the relatively weak transistors in the cell itself, so that they can easily override the previous state of the cross-coupled inverters. Careful sizing of the transistors in an SRAM cell is needed to ensure proper operation.

Simulation results show that by using bitline segmenting architecture (Figure 4.4(a) and 4.4(b)), bitline swing as low as $V_{dd}/4$ can result in a successful *write* operation under the worst case offset condition between the cell inverters. It is evident that unlike conventional *write* operation, the neighboring bitlines are not discharged; this adds to the power saving capability of the scheme.

5.3 Access-time

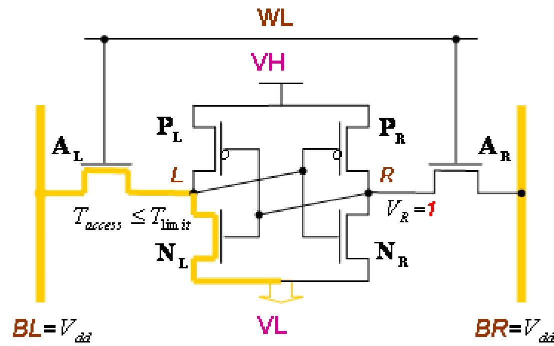


Figure 5.3: 6T *access* operation.

The cell *access-time* (T_{access}) is defined as the time required to produce a pre-specified voltage difference ($\triangle MIN \approx 0.1V_{dd}$) between two bit-lines (bit-differential between BL and BR). As shown in Figure 5.3, T_{access} must not exceed the maximum allowable discharge time (T_{limit}). The sizes of the access transistor (i.e.: A_L) and pull-down transistor (i.e.: N_L) influence

the discharge time, and therefore the speed of SRAM.

5.4 Hold

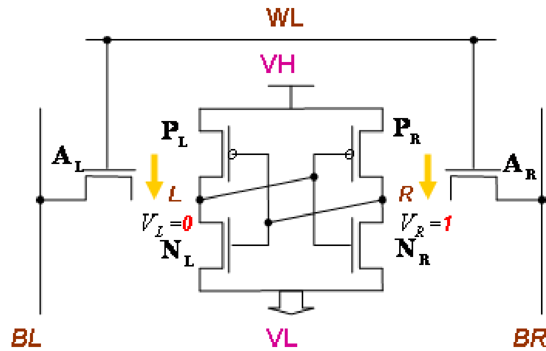


Figure 5.4: 6T *hold* operation.

Looking at Figure 5.4, if the wordline is not asserted, the access transistors (A_L and A_R) disconnect the cell from the bitlines. The two cross coupled inverters formed by $N_L - P_L$ and $N_R - P_R$ will continue to reinforce each other as long as they are connected to the power supply.

In *hold* (standby) mode, all transistors are in the weak inversion region. That is, when the wordline is activated to access a desired cell, the cells on the same row and non-selected columns are kept in the nominal voltage condition. Therefore, these cells go into the *accessed retention* mode. As mentioned earlier, a cell cannot discharge its corresponding bitline if it goes to the *hold* mode (standby or *accessed retention* mode). Thus, the power consumption is reduced compared to the power consumption of SRAMs not using bitline and wordline segmenting architecture. In addition, the configuration of having the bits of each word of multiple interleaved words scattered with an equal distance from each other on each row

has the additional benefit of higher tolerance to soft errors compared to low voltage, single or multiple non-interleaved words per row scenarios. Distributing the bits of a word over the row reduces the number of soft errors caused by a single radiation event.

Part III

SRAM Design Considerations and Analysis

As we go to increasingly smaller geometries (i.e. smaller nodes) in order to achieve higher integration and lower cost, variation impacts are becoming more critical in the design of SRAM technologies. For example, in the 65-nm node, we can (arguably) avoid the variation effects by guard-banding the design and following all the prior flows in design development. However, in smaller geometries such as the 22-nm or 16-nm nodes, we are required to perform careful variation effect analysis by understanding the potential impacts of different types of variation on our design. A recent report by SOlido shows that 65% of engineers surveyed see variation effects as their top concern for analysis in the next 2 years [48].

There are many types of variation to consider. These variations can be classified into three groups: *Operational*, *Fabrication*, and *Implementation*, as shown in Figure III-A.

1. ***Operational:*** This includes *environmental* and *loading* variations—which are more the effects of variation around the design—such as the voltage of the power supply (V), temperature (T), and different loading conditions. For example, in the design of SRAM, there may be different conditions in the actual implementation which can make them function differently than intended. The environmental fatigue phenomena (HCI, NBTI etc.) are examples of temporal variations that could also be placed in this category.

2. ***Fabrication:*** This involves *global* and *local* process variations (PV). *Global* variations have been historically analyzed through corner-based models, but now, as we go to smaller geometries, the *local* effects are starting to be almost as important as the *global* effects. Therefore, they have to be considered as well when using Monte Carlo based tools.

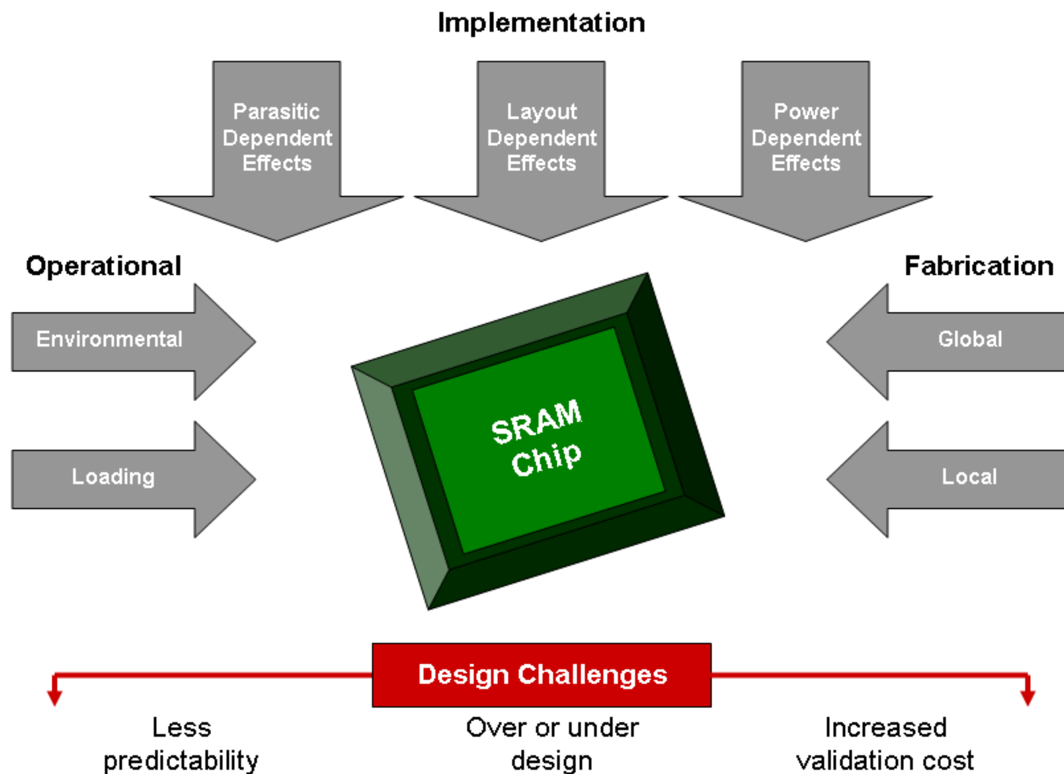


Figure III-A: Classification of variations in IC Design.

3. ***Implementation:*** This includes layout-based variation effects. Physical *parasitic effects* have been one of the design challenges during the last two decades. Similarly, *power integrity connectivity effects* for supply demand has increasingly become a concern in the last few years. More recently, the concern has been *layout dependent effects*, which describes the change in electrical characteristics (such as V_{th} and effective length (L_{eff})) of specific devices depending on where they are placed within the SRAM.

All these variations are a huge challenge that designers have to face. Designers often have to make a choice between running fewer simulations in order to meet a deadline, which

means there is *less predictability* in the quality of the design, or they can do more analysis and run the risk of *increased validation cost*. This represents the fundamental challenge for today‘s designers: choosing between *over* or *under designing*.

As briefly pointed out in Section 4.1, the challenge of cell design is due to conflicting interactions between some main factors including, but not limited to: 1) Minimizing the cell area to achieve high density memory, reduce power, and reduce the cost of the chip. 2) Maintaining cell stability with minimum voltage to prevent yield loss due to data corruption (see Section 6.2). 3) Good soft error immunity—in systems with a high reliability requirement, a data error due to a soft error can cause catastrophic failures (see Section 6.3). 4) High cell read current to minimize access-time. 5) Minimum word line pulse width to conserve power (by reducing bitline swing). 6) Low leakage current, especially for battery operated systems [111].

For example, to maintain cell stability and good soft-error immunity (transient errors such as those induced by radiation) [17] while keeping access-time short, one might specify large transistor sizes. Unfortunately, large transistors occupy more area and result in increased leakage. Similarly, improving static noise margin (SNM) with smaller pass transistors can lead to a worse write margin [111]. Transistor sizing and circuit styles for 6T-SRAM components (decoders, sense amps, etc.)—and the interconnect sizing, buffers, and SRAM array partitioning—must all be balanced with considerations to the delay, area, and power consumption.

For our transistor sizing we have used the following formula from J. Rabaey [141], with some adjustments to improve performance while lowering leakage power:

$$\Delta V = \left\{ V_{DSATn} + CR(V_{\mathrm{dd}} - V_{Tn}) - \sqrt{V_{DSATn}^2(1+CR) + CR^2(V_{\mathrm{dd}} - V_{Tn})^2} \right\} / CR$$

where, CR is called the cell ratio and is defined as:

$$CR = (WNL/LNL)/(WAL/LAL)$$

where WAL and LAL are the width and length of A_L, respectively, and WNL and LNL are the width and length of N_L in Figure III-B.

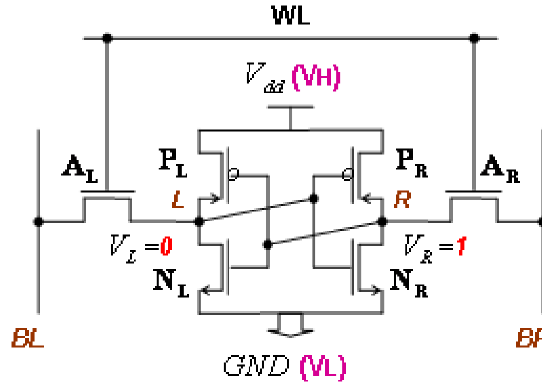


Figure III-B: 6 transistor (6T) storage cell (repeated for convenience).

In addition to the increase in the number of variation types, several reliability issues such as NBTI and HCI are also becoming critical concerns in newer technology nodes. Advances in semiconductor manufacturing techniques and ever increasing demand for faster and more complex integrated circuits (ICs) have driven the associated Metal-Oxide-Semiconductor field-effect transistor (MOSFET) to scale to smaller dimensions. However, it has not been possible to proportionately scale the supply voltage used to operate these ICs due to factors such as compatibility with previous generation circuits, noise margin, power and delay requirements, non-scaling of the threshold voltage, subthreshold slope, and parasitic capacitance. As a result, internal electric fields increase in aggressively scaled MOSFETs, which comes with the benefit of increased carrier velocities (up to velocity saturation), and hence increased switching speed [51]. However, it also presents a major reliability problem for the long term operation of

these devices, as high fields induce hot carrier injection which affects device reliability.

Large electric fields in MOSFETs imply the presence of high-energy carriers, referred to as **"hot carriers."** These hot carriers have high enough energy and momentum to be injected from the semiconductor into the surrounding dielectric film such as the gate and sidewall oxides, as well as the buried oxide in the case of silicon on insulator (SOI) MOSFETs.

The presence of such mobile carriers in the oxide triggers numerous physical damage processes that can drastically change the device characteristics over prolonged periods. The accumulation of damage can eventually cause the circuit to fail as key parameters such as threshold voltage can be modified. The accumulation of damage in the device due to hot carrier injection is called **"hot carrier degradation."**

The useful lifetime of circuits and integrated circuits based on such a MOS device are thus affected by the life-time of the MOS device itself. To assure that integrated circuits manufactured with these minimal geometry devices will not have their useful life impaired, the lifetime of the component MOS device must have its NBTI and HCI degradation well understood. Failure to accurately characterize NBTI and HCI lifetime effects can ultimately affect business expenses such as warranty and support costs and impact marketing and sales promises for a foundry or IC manufacturer.

Factors impacting the stability, robustness, and reliability of SRAM are explained in further details in chapters 6 and 7. We model the impact of process and operation variations on the performance of SRAM in Chapter 9.

Chapter 6

Design Considerations and Analysis, Device

6.1 D2D and WID variations

Process variations can be classified as systematic or random where *systematic variation* is deterministic in nature and is caused by the structure of a particular gate and its topological environment. For instance, wire thicknesses will polish differently during Chemical Processes and Materials (CPM) depending on the density of the surrounding routing. Also, poly gate width has a deterministic dependence on the spacing of neighboring poly lines due to limitations of the lithography and the application of optical proximity correction (OPC) methods. *Random variations* are unpredictable in nature and include random variation in the device length, oxide thickness, and discreet doping fluctuations. Analysis of the impact of deterministic variations on circuit delay is relatively straightforward, given accurate models of their dependence on physical topologies and the needed layout information. Methods have been proposed to include deterministic device length variations [126] and interconnect variations [107] in the

analysis of circuit performance. However, often the necessary models and layout information for incorporating deterministic variations in delay computation are not available and hence, deterministic variations are treated as random variations. The *random* and *systematic* variations in fabrication (process), operation, and implementation parameters have emerged as a major challenge in circuit design in the nanometer regime [115, 121, 27].

Process variations can be further classified as either D2D or WID variation. D2D variations describe fluctuations that occur from one die to the next, meaning that the same device on a chip has different features among different dies of one wafer, from wafer to wafer, and from wafer lot to wafer lot, as discussed further in the following paragraphs. WID variations describe fluctuations in device features that are present within a single chip, meaning that a device feature varies between different locations on the same die. Often, intra-chip variations exhibit spatial correlations, where devices that are close to each other have a higher probability of being alike than devices that are placed far apart. WID variations also exhibit structural correlations, meaning that devices that are structurally similar have an increased likelihood of having similar device features. For instance, devices oriented in the same direction tend to be more alike. With increased process scaling, WID variations have become a more dominant portion of the overall variability, meaning that devices on the same die can no longer be treated as identical copies. In our modeling, discussed in Chapter 9, we are concerned with the impact of systematic D2D and both systematic and random WID variations on circuit performance.

The process sources of the inter-die (D2D) and the intra-die (WID) variation includes variation in channel length, channel width, oxide thickness, threshold voltage, line-edge roughness, and *random* dopant fluctuations (the *random* variation in the number and location of

dopant atoms in the channel region of the device results in the *random* variation of the transistor threshold voltage (RDF)) [115, 121, 27, 21]. The *operational* sources of D2D and WID variation include variation in loading and environmental conditions (such as supply voltage and temperature) [197]. The *implementation* sources of D2D and WID variation includes parasitic (interconnects) and layout-based parameters—such as back-end-of-line (BEOL) and depth of focus (DOF) during lithography [197]. These different sources of variation result in significant differences in the delay and the leakage of digital circuits [115, 121, 27, 21]. The D2D variation in a parameter (say threshold voltage (V_{th})) modifies the value of that parameter in all the transistors within a die in the same direction (i.e., the threshold voltage of all the transistors either increase or decrease). This principally results in a spread of the delay and the leakage, but does not cause a mismatch between different transistors in a die. On the other hand, WID variation shifts the process parameters of different transistors within a die in different directions (e.g., V_{th} of some transistors will increase whereas others will decrease).

The WID (or intra-die) variation can be *systematic* (i.e., parameter change of one transistor depends on the parameter change of a neighboring transistor) or *random* (i.e., parameter differences of two neighboring transistors are completely independent). An example of *systematic* WID variation is the change in the transistor channel length across a die that is spatially correlated. The RDF induced V_{th} variation is a classic example of the *random* WID variation. The *systematic* variation does not result in large differences between the two transistors that are in close spatial proximity. The *random* component of the WID variation can result in a significant mismatch between the neighboring transistors in a die) [115, 121, 27, 21]. In an SRAM cell, a mismatch in the strength between the neighboring transistors, caused by WID variation,

can result in the failure of the cell [115, 113], as is explained in Chapter 8.

Among the different sources of *random* WID variation, the most significant one is the threshold-voltage (V_{th}) variation due to RDF. The impact of the *random* dopant effect is most pronounced in minimum-geometry transistors commonly used in area-constrained circuits such as SRAM cells [115]. However, the impact of WID channel length variation and, to a lesser extent, the impact of V_{dd} and temperature variation, is also becoming more critical as we move towards the 16-nm technology node. Hence this thesis considers, among others, not only the variation of V_{th} but also the variation of channel length, temperature, and V_{dd} in its variability analysis. We have also considered the correlation among the threshold voltage, channel length, and V_{dd} of different transistors in a cell to better understand the impact of the *systematic* variations.

The parametric variation, and in particular the V_{th} fluctuation due to RDF, is a strong function of the size of different transistors in the cell (channel length (L), width (W). Hence, the failure probability of SRAM can be reduced by optimally designing the size of different transistors. However, any such optimization has to consider its impact on the overall area and leakage of the SRAM array. Moreover, the memory organization (i.e., number of rows, number of columns, and the number of redundant columns) also has a strong impact on the memory-failure probability. Hence, a hybrid analytical-empirical modeling of the SRAM cell and architecture is very important to reduce the memory-failure probability and to improve the yield in nano-scaled SRAM. The discussion of Failures in SRAM is presented in Chapter 8.

6.2 Static Noise Margin (SNM)

As discussed previously, for *static noise margin (SNM)* analysis, we classify manufacturing variations as *systematic* or *random*. *Systematic* variations are predictable in nature and depend on deterministic factors such as layout structure and the surrounding topological environment [5, 127]. On the other hand, *random* variations are unpredictable and are caused by *random* uncertainties in the fabrication process such as microscopic fluctuations in the number and location of dopant atoms in the channel region [5]. *Random* variations are harder to characterize and can have a detrimental effect on the yield of critical modules in a circuit.

Random variations can cause a significant mismatch in neighboring devices and hence are largely responsible for the poor yield of *SRAM* arrays in scaled technologies [40, 5]. SRAM yield is very important from an economic viewpoint due to the critical and the ubiquitous nature of memory in modern processors and SoCs. Density is a very important metric for memory and hence SRAM cells use the smallest manufacturable device sizes in a given technology. However, the threshold voltage variation due to *random dopant fluctuation* is inversely proportional to gate area [134, 5]. Due to this dependence, the nanoscale transistors in a memory cell see a highly pronounced *random* dopant effect. Moreover, SRAM cells are traditionally designed to ensure that the contents of the cell are not altered during *read* access while the cell should be able to quickly change its state during the *write* operation. These conflicting *read* and *write* requirements are satisfied by balancing the relative strengths of the devices in the design. Such careful design of an SRAM cell provides stable *read* and *write* operation, but it also makes the cell vulnerable to the failures caused by *random* variation in the device strengths.

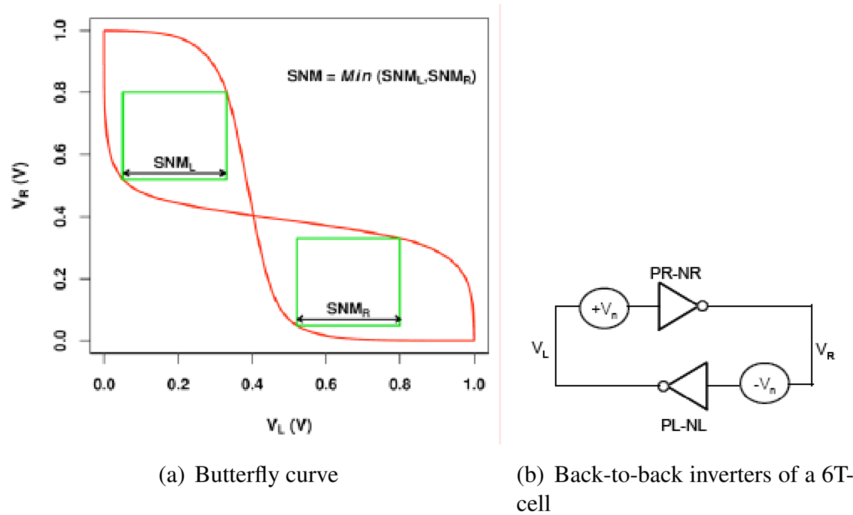
Due to increased sensitivity of SRAM designs to process variation, failure analysis of a memory cell has become an extremely important exercise. The electrical yield of a cell is typically analyzed through Monte-Carlo simulations which treat the threshold voltage of each device in the cell as an independent *random* variable. However, the large number of simulations required to obtain full stability coverage solely through Monte Carlo simulations makes them computationally prohibitive. R. Heald et al. [63] illustrate this problem with an example of a 4 MB cache. The authors show that a typical 4 MB cache with *error correcting code (ECC)* cells contains approximately 38 million cells. To ensure that there is at most one failure in this cache, the circuit must operate correctly up to 5.44 sigmas. This sort of fault coverage can only be verified by millions of simulations. Furthermore, Monte-Carlo simulations provide little insight to the designer about optimizing cell yield in subsequent iterations. A modeling based approach, on the other hand, can not only be used to estimate cell failure probability in an efficient manner, but it can also guide cell and architecture optimization for yield enhancement.

Analytical modeling of SRAM cell stability is not an entirely new concept. Earlier work in this field focused on characterizing SRAM robustness by modeling the *SNM* of the cross-coupled inverters using a graphical technique [152, 20]. More recently, there have been efforts in characterizing cell stability during *read* and *write* operations [115, 72]. Most of these works rely on device equations to solve for parameters such as *SNM*, *read disturbance* and *inverter trip-point*.

For our *SNM* analysis, we have adopted a recent simple and accurate method for modeling *read*, *write*, and *access* failure probabilities of an SRAM cell introduced by K. Agarwal et al. [5]. After evaluating each of the *read*, *write*, and *access* failure probabilities using this

model, we validate our results by comparing them to the graphical method (also known as butterfly curve method, which is used by Tanmy [154] and is explained later in this section). At the end of this section, we will demonstrate (using Mukhopadhyay [115] statistical analysis modeling results) that, although insightful, the Agarwal *SNM* modeling (or graphical method) alone is not sufficient to show the cumulative effects of all three failures on the yield, and consequently justify why this thesis proposes the new model VAR-TX. But first, we will define noise margin and *SNM*, widely used in the VLSI field.

In electrical engineering, ***noise margin*** is the amount by which a signal exceeds the minimum amount for proper operation [188]. The noise margin of an SRAM cell is defined as the minimum amount of DC noise required to flip the state of the cell [5]. However, as shown in Figure 6.1, this noise metric, called ***Static Noise Margin (SNM)***, assumes that the two storage nodes in the cell are subject to equal and opposite DC noise offsets [152, 20].



(a) Butterfly curve (b) Back-to-back inverters of a 6T-cell

Figure 6.1: Graphical method of characterizing *Static Noise Margin (SNM)* of an SRAM cell [5].

The *SNM* of a cell is often used as a measure of the robustness of an SRAM cell

against flipping [20]. It represents the resilience of the design in the event of a disturbance. Traditionally, *SNM* is extracted from the butterfly curve in the following way (Figure 6.1(a)): First, the largest squares that can be inscribed in the two openings of the butterfly curve are found. Then, the *SNM*is defined as the length of the side of the smaller of the two squares. The butterfly curve is formed by overlapping the two Voltage Transfer Characteristics (VTC) of nodes V_L and V_R, shown in Figure 6.1(b). In Figure 6.1(b) PL-NL and PR-NR represent the left and right inverters (as we saw earlier in Figure 4.1), respectively. The butterfly curve is typically symmetric which makes the size of the two squares inscribed within the openings the same.

Alternatively, the SRAM noise margin can be characterized by modeling the cross-coupled inverters as a positive feedback loop system. In their work, J. Lohstroh et al. [100] show that the 6T-cell is on the verge of instability if its loop gain is unity. K. Agarwal et al. [5] generalize the loop gain concept and propose a new criterion for quantifying 6T-cell stability in the presence of DC noise offsets.

These authors [5] provide a theoretical framework for computing DC noise margins and demonstrate that the noise margin is better characterized by computing the loop gain of the cross-coupled inverters in the memory cell. They apply the loop-gain concept to the *read stability* problem and develop a *read stability* metric called *read noise margin (RNM)*. Subsequently, they show that the *RNM* has a Gaussian distribution and can be easily modeled as a linear function of the *random* parameter variations of different transistors in the cell. They also make the observation that *write* failures occur due to timing violations. Furthermore, they show that the inverse of the *write* and *access* delays also follow Gaussian distributions and can be

characterized by sensitivity-based linear models.

To quantify cell stability in the presence of DC noise offsets, we adapt Agarwal's method and begin by considering the case of an SRAM cell which stores a value ($V_L = 0$ and $V_R = 1$). Let us assume that a DC noise disturbance at node L causes its potential V_L to rise above zero. Our objective is to find the minimum DC noise disturbance at node L that causes the cell to lose its state. Let us assume that the DC transfer characteristics of the $PR-NR$ and the $PL-NL$ inverters (labeled in Figure 6.1(b)) can be modeled by functions f and g respectively. For a symmetric cell, the two functions should be identical, but they will differ due to *random* mismatches in the device characteristics.

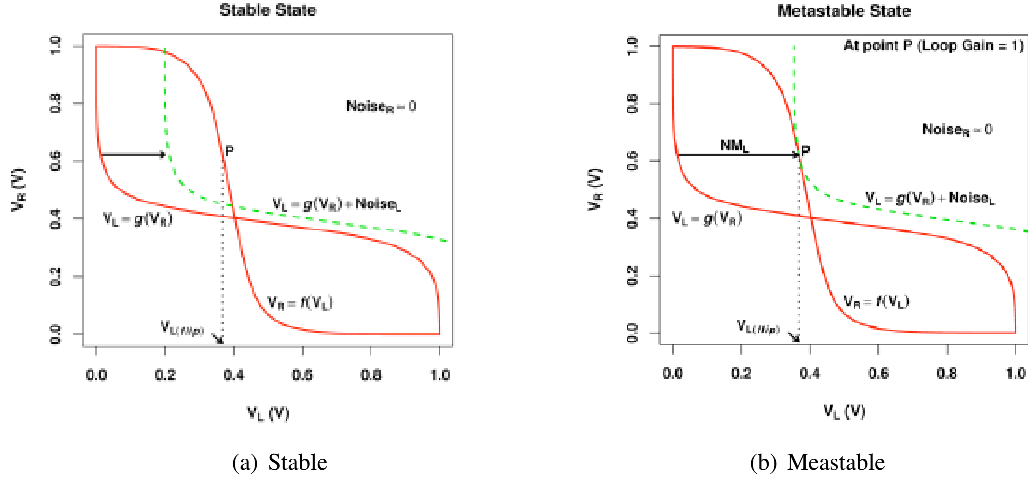
$$\begin{aligned} V_R &= f(V_L) \qquad (InverterPR-NR) \\ V_L &= g(V_R) \qquad (InverterPL-NL) \end{aligned} \tag{6.1}$$

Due to the non-linear nature of the transfer-characteristics f and g, the gains of the two inverter stages depend on their input voltages. Hence, a disturbance at node L causes a change in the gain of the $PR-NR$ stage. A noise offset at node L also changes the potential at node R and thus impacts the gain of the feedback stage. The loop gain of the system as a function of the node L potential can be expressed as:

$$LoopGain(V_L) = \frac{\partial f}{\partial V_L} \cdot \frac{\partial g}{\partial V_R}|_{V_R = f(V_L)} \tag{6.2}$$

The value of V_L that causes the loop gain to become unity is denoted by $V_{L(flip)}$. This value, as shown in Figure 6.2, represents the minimum DC potential required to flip the

contents of a cell. In other words, $V_{L(flip)}$ is the maximum potential that can be tolerated by node L without altering its state from zero to one.



(a) Stable (b) Meastable

Figure 6.2: (a) Stable and (b) *metastable* states of an SRAM cell in the presence of a positive DC noise offset ($Noise_{\mathrm{L}}$) on node L [5].

Figure 6.2 shows the significance of $V_{L(flip)}$ in analyzing the noise margin of an SRAM cell. The figure shows the butterfly curves of the cross-coupled inverters as modeled by Equation (6.1). The figure also shows the shifted DC transfer characteristics $V_L = g(V_R) + Noise_{\mathrm{L}}$ of the feedback inverter due to a positive DC noise offset at the node $L(Noise_{\mathrm{L}})$. For a small noise offset, the cell maintains its state because it has a stable operating point in the vicinity of the initial state ($V_L = 0$ and $V_R = 1$). However, as the noise is increased, the shifted curve will move until it intersects the forward inverter characteristics at only one point (as labeled P in Figure 6.2).

If the noise is increased beyond this point, then the cell will lose its state because the two DC characteristics will not have a stable intersection point required to maintain the initial state. The state of the cell, when the two curves barely touch each other and the cell is on the

verge of instability, is defined as the *metastable* state. Interestingly, the potential at node L in the *metastable* state is $V_{L(flip)}$. Based on this observation, we define the noise margin of the cell as:

$$NM_L = V_{L(flip)} - g[f(V_{L(flip)})] \tag{6.3}$$

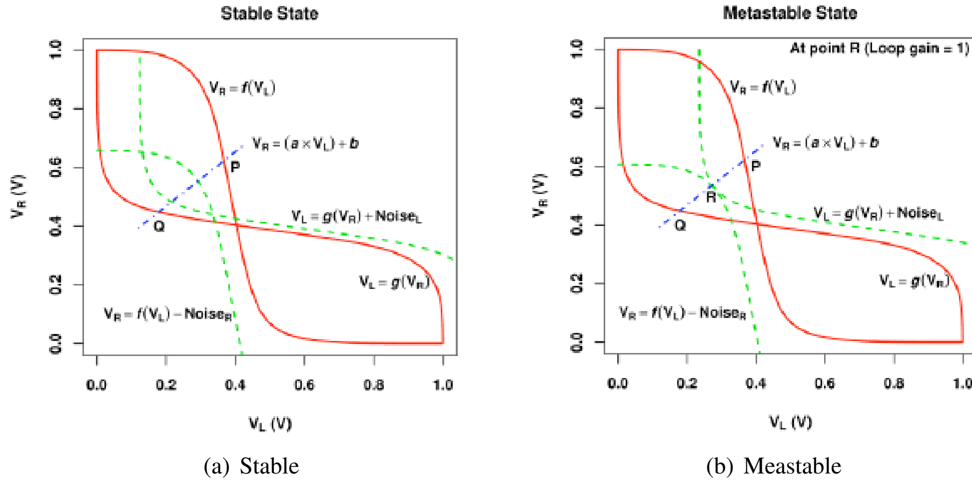
Next, we consider the case when the cell still stores a value ($V_L = 0$ and $V_R = 1$) but a negative DC noise disturbance is applied at node R. The negative disturbance causes node *R's* potential to fall below one. Similar to Equation (6.2), the loop gain of the system as a function of node R potential (V_R) can be expressed as:

$$LoopGain(V_R) = \frac{\partial f}{\partial V_L}|_{V_L=g(V_R)} \cdot \frac{\partial g}{\partial V_R} \tag{6.4}$$

Now we can compute the potential (V_R) that causes the loop gain to become unity ($V_{R(flip)}$) by solving the above equation. As shown in the Figure 6.3, a negative offset at node R shifts the DC transfer characteristics of the forward inverter vertically by $Noise_{\mathrm{R}}$. Similar to Equation (6.3), the noise margin of the cell from this side can be calculated by:

$$NM_R = f[g(V_{R(flip)}] - V_{R(flip)} \tag{6.5}$$

NML models the maximum positive noise a cell can tolerate at its zero node without losing its contents, while *NMR* represents the maximum tolerable negative noise offset at the



(a) Stable (b) Meastable

Figure 6.3: (a) Stable and (b) *metastable* states of an SRAM cell when a positive DC noise offset ($Noise_L$) is applied on node *L* and negative noise ($Noise_R$) is applied on node *R* [5].

node storing one. Given *NML* and *NMR*, the noise margin of a cell can be expressed as:

$$NM = Min(NM_L, NM_R) \tag{6.6}$$

The above noise margin metric is different from *SNM* because it characterizes a cell's stability under the assumption that only one side of the cell is disturbed by external noise. *SNM*, on the other hand, is a measure of noise margin when simultaneous positive and negative DC noise offsets are present at the two nodes of the cell. SRAM failures due to *read noise* and also *alpha particle* strikes usually occur due to one-sided disturbances. Therefore, the noise margin model of Equation (6.6) is more useful in checking cell stability in the presence of DC noise offsets. However, for the sake of completeness of analysis, the above loop-gain concept can be extended to model cell stability in the presence of noise disturbances at both ends of a cell.

To find the SRAM stability criterion in the presence of double-sided noise, once again

we consider the case of an SRAM cell storing a value ($V_L = 0$ and $V_R = 1$). However, this time we assume that both nodes in the cell are subjected to external disturbances. When V_L and V_R can take any possible value, the loop gain of the system as a function of the potentials of node L and node R can be expressed as:

$$LoopGain(V_L, V_R) = \frac{\partial f}{\partial V_L} \cdot \frac{\partial g}{\partial V_R} \tag{6.7}$$

Figure 6.3 demonstrates the stable and the *metastable* state of a cell in the presence of double-sided noise. The figure shows that noise offsets at nodes L and R result in a horizontal and vertical shift in the transfer characteristics of the two inverters. In the *metastable* state, the combination of $Noise_{\mathrm{L}}$ and $Noise_{\mathrm{R}}$ shifts the butterfly curves such that they intersect at only one point (point R). In this state, the cell is on the verge of instability as a small amount of additional noise will change the value of the cell.

In the case of SRAM, three types of noise margins need to be evaluated: *hold noise margin*, *read noise margin*, and *write noise margin*. For both simplicity and exploration reasons, we will show the simpler method of evaluating the three noise margins used by T.A. Shah in his work [154].

6.2.1 Hold Noise Margin

When the 6T-cell is in the IDLE condition (wordline WL = 0 and bitlines BL=BR= 1), the noise margin is evaluated using the procedure mentioned by Richard E. et al. [162]. First, the feedback loop in the cross-coupled inverters is broken. WL is kept LOW and BL is kept

HIGH. The voltage at V_L is swept from 0 to V_{dd} and the voltage at V_R is measured and the corresponding VTC is plotted as Real(y) vs. Real(x). The same VTC is plotted again, but with the axes switched. The size of the resulting two squares inscribed within the butterfly curve turn out to be almost the same—which translates into an *SNM* of about 370mV for the *hold* mode (for the 45-nm node).

6.2.2 Read Noise Margin

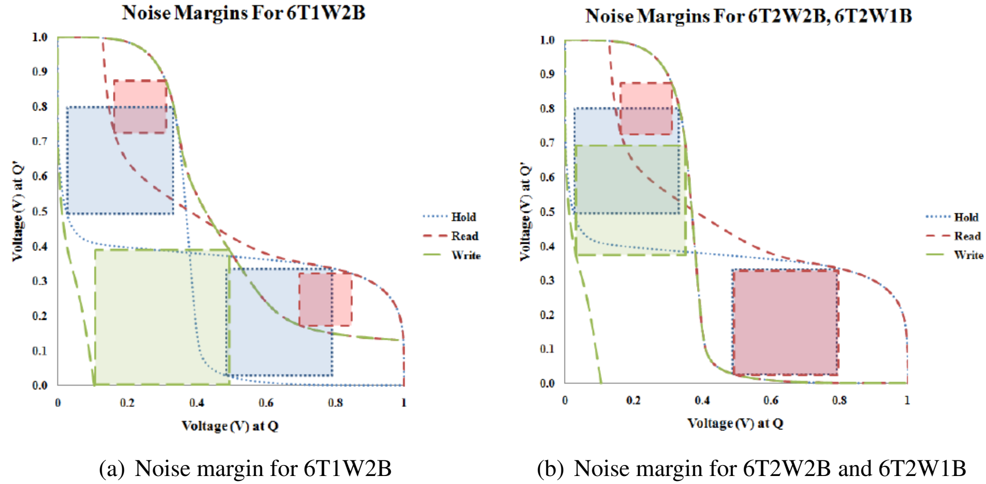
Read noise margin is measured using the same procedure as the *hold noise margin*, but the wordline is held HIGH. All other conditions are the same as for the *hold noise margin*. The *SNM* for the *read* mode turns out to be around 153mV (for 45-nm)—which is less than half of both the *hold* and *write* modes.

6.2.3 Write Noise Margin

For the *write noise margin* measurement, two VTCs are plotted. For the first VTC, the bitline BL is kept HIGH and for the second VTC, BL is kept LOW, while pulsing/activating WL as described by J. Wang [179]. The butterfly curve is then obtained to measure the *SNM* for the *write* mode—which turns out to be around 406mV (45-nm).

Figure 6.4(a) and 6.4(b) illustrate how *hold noise margin*, *read noise margin*, and *write noise margin* are different for different SRAM architectures [180]. In Figure 6.4(a) and 6.4(b), M.C. Wang [180] shows that, as compared to a standard 6T-SRAM, designs with dual wordlines (6T2W2B, 6T2W1B) suffer a 17% reduction in the *write* margin, but gain 103% more *read noise margin* for reading a 0. Since the "*read noise margin*" typically has the most

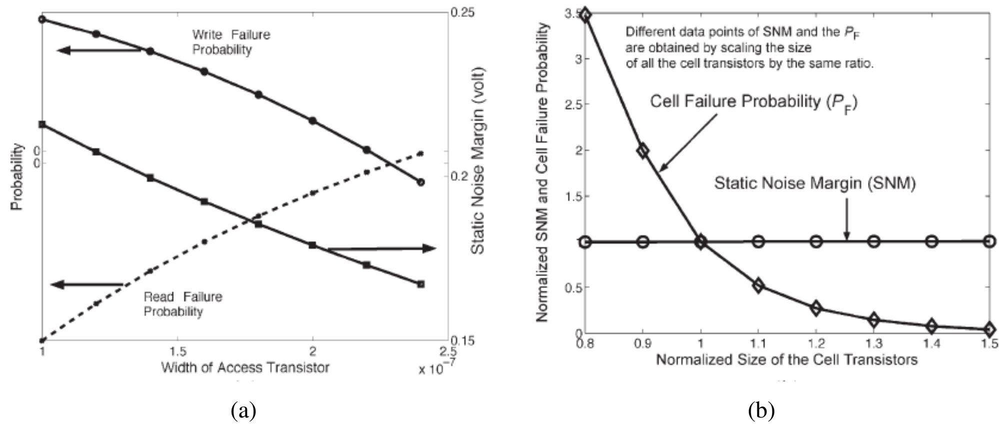
adverse impact on the stability of SRAM, such a trade off (a *read noise margin* improvement of 103% at the cost of only 17% decline in the *write noise margin*) should be considered favorable.



(a) Noise margin for 6T1W2B (b) Noise margin for 6T2W2B and 6T2W1B

Figure 6.4: Comparison of *hold noise margin*, *read noise margin*, and *write noise margin* of 6T-SRAM designs with (a) single wordline (6T1W2B) and (b) dual wordlines (6T2W2B, 6T2W1B). Q and Q' represent the left-node (*L*) and right-node (*R*) of the 6T-cell [180].

Considering the discussion and plots above, we can conclude that it is desirable to have a sufficiently larger noise margin to ensure that flipping does not occur. However, as pointed out earlier, an increase in *SNM* makes the cell difficult to *write* by increasing its data-holding capability, which increases *write* failures. This means that, although the *SNM* can be increased by careful sizing, the cumulative/joint failure probability of SRAM is not reduced correspondingly. For example, reducing the size of the access transistors (i.e. reducing the widths of A_L and A_R in Figure 4.1) improves the *SNM* [21, 115] and therefore, decreases read-failure probability. At the same time, however the *write*-failure probability increases (Figure 6.5(a)). Hence, the reduction in the sizes of access transistors that results in a maximum *SNM* does not necessarily correspond to a minimum-failure probability (Figure 6.5(a)). Moreover, increasing



(a) (b)

Figure 6.5: Variation of SNM and failure probability with (a) width of the access transistors; and (b) normalized cell area [115].

the size of all the transistors in a cell by the same factor does not modify the *SNM*. However, an increase in the size of all the transistors in a cell considerably reduces its failure probability by reducing the standard deviation of the V_{th} variation (Figure 6.5(b)). Using the proposed models, it is observed that *SNM* does not have a strong relationship with the parametric failure of the memory.

Consequently, an increase in the *SNM* does not necessarily reduce the overall failure probability and an *SNM*-based analysis of the cell does not directly correspond to the memory failure probability and yield. Hence, a statistical analysis and design of the cells and memory architecture is necessary to ensure acceptable yield in the nanometer regime. This thesis' proposed model VAR-TX (explained in Chapter 9) and its simulation results (illustrated in Chapter 10) provide such a statistical analysis and incorporates the design considerations that are described in this chapter.

6.3 Soft Error

In electronics and computing, a **soft error** is a signal or datum that is wrong. Soft errors involve changes to data but not to changes in the physical circuit itself [49]. That is, soft errors are the change of electrons, but not the changes of atoms. Therefore, a soft error may be corrected by rewriting the data where it was lost with no adverse effects to the circuit. Soft errors can occur on transmission lines, in digital logic, analog circuits, magnetic storage, and elsewhere, but are most commonly known in semiconductor storage, namely SRAM. The main causes of soft errors are package radioactive decay (usually due to alpha particle emission) and cosmic rays creating energetic neutrons and protons [198].

In SRAM array design, multiple words can be placed on a single row with the bits of each word either next to each other (non-interleaved) or equally scattered at a certain distance (interleaved). The non-interleaved design has the advantage of less architectural complexity, but it has the disadvantage of having a higher chance of experiencing soft errors [106]. Therefore, since distributing the bits of a word over the row reduces the number of soft errors caused by a single radiation event, our SRAM array uses an interleaved design.

6.4 Negative Bias Temperature Instability (NBTI)

NBTI Overview:

The rapid scaling of CMOS technology has resulted in new reliability concerns, such as negative bias temperature instability (NBTI), and non-conductive stress (NCS), among others [138, 104, 132, 184]. NBTI has become the primary limiting factor of circuit lifetime. NBTI

primarily affects pMOS devices, since they almost always operate with negative gate-to-source voltage; however, the very same mechanism also affects n-channel MOS (nMOS) transistors when biased in the accumulation regime, i.e. with a negative bias applied to the gate. NBTI manifests itself as an increase in the threshold voltage (V_{th}) and a consequent decrease in the drain current and transconductance (gm), the ratio of the current change at the output port to the voltage change at the input port; $gm = \frac{\Delta I_{\text{out}}}{\Delta V_{\text{in}}}$. The degradation exhibits logarithmic dependence on time [151].

The NBTI effect can be physically described by reaction-diffusion (R-D) theory as a continuous generation of charges at the Si-SiO interface. In the reaction phase, some Si-H bonds at the Si-SiO interface are broken under the vertical electrical stress. This phenomenon results in the generation of interface charges [8, 177, 19]. Given the initial concentration of the Si-H bonds (N_0) and the inversion carriers (P), the generation rate of the interface traps N_{IT} can be calculated [184]. D. K. Schroder et al. [151] point out that, in sub-micrometer devices, nitrogen is incorporated into the silicon gate oxide to reduce the gate leakage current density and prevent the boron penetration. However, incorporating nitrogen enhances NBTI. For newer technologies (32-nm and below), high-K (HK) metal gate stacks are used as an alternative to improve the gate current density for a given equivalent oxide thickness (EOT). Even with the introduction of new materials like hafnium oxides, NBTI remains. D. Schroder et al. [151] speculate that it is possible that the interfacial layer, which is composed of nitrided silicon dioxide, is responsible for those instabilities. This interfacial layer results from the spontaneous oxidation of the silicon substrate when the HK is deposited. To limit this oxidation, the silicon interface is saturated with N resulting in a very thin and nitrided oxide layer.

It is commonly accepted that two kinds of trap contribute to NBTI [151]:

- First, ***interface traps*** are generated. These traps cannot be recovered over a reasonable time of operation. Some refer to them as *permanent traps*. These traps are the same as the ones created by Channel Hot Carrier. In the case of NBTI, it is believed that the electric field is able to break the Si-H bonds located at the Silicon-oxide interface. H is released in the substrate where it migrates. The remaining dangling Si- bond (Pb center) contributes to the threshold voltage degradation.

- In addition to the generated interface states, some ***pre-existing traps*** are located in the bulk of the dielectric (and are supposedly nitrogen related) and are filled with holes coming from the channel of the pMOS. These traps can be emptied when the stress voltage is removed. This V_{th} degradation can be recovered over time.

The existence of two coexisting mechanisms for NBTI has created a large controversy, with the main issue being the recoverable aspect of interface traps. Some suggested that only interface traps were generated and recovered; today this hypothesis is ruled out. The situation now is clearer, but not completely solved. Some suggest that interface trap generation is responsible for hole trapping in the bulk of the dielectric. A tight coupling between the two mechanisms may exist, but nothing has been demonstrated clearly [150].

The degradation due to NBTI may result in up to 50 mV shifts in the threshold voltage (V_{th}) throughout the lifetime of a circuit, which translates to more than a 20% degradation in the circuit speed, or in extreme cases, to a functional failure [26, 151]. Experimental data further indicates that NBTI worsens exponentially with thinner gate oxide and higher operating

temperature (T) [105, 151, 89]. In fact, as the gate oxide becomes thinner than 4 nm (as in nodes below 32-nm), NBTI has gradually become the dominant factor to limit circuit lifetime [74, 184].

With the introduction of High-K Metal gates, a new degradation mechanism, Positive Bias Temperature Instabilities (PBTI), has appeared. The PBTI affects the NMOS transistor when positively biased [8]. Since, in this particular case, no interface states are generated and 100% of the V_{th} degradation may be recovered, the impact of PBTI is not as severe as that of NBTI.

In short, NBTI manifests itself as an increase in $|V_{\text{th}}|$, and consequently, an increase in logic delay, whenever a PMOS transistor is under stress ($|Vgs| > |V_{\text{th}}|$). Relaxation of the stress (Vgs = 0) can recover only part of the V_{th} degradation [7], causing an overall increase in delay over time (*NBTI degradation*). If not appropriately provisioned for, increased delay can result in timing failures on critical logic paths. NBTI degradation is frequency independent [7, 177] but increases with supply voltage (V_{dd}) and temperature [34].

Even though tremendous efforts have been spent to improve the fabrication process, the impact of NBTI on circuit performance has become so severe that technology improvement alone is not sufficient, especially after the introduction of high-k gate dielectrics. For nanoscale CMOS circuits, it is essential to develop design methods to understand, simulate, and minimize the degradation of circuit performance in the presence of NBTI, in order to ensure reliable circuit operation over a desired period of time.

Traditionally, guardbanding has been used to protect against NBTI. For example, the operating frequency can be reduced or the supply voltage can be increased to offset the

degradation over the lifetime of a design. Unfortunately, guardbanding incurs a throughput or power cost over the entire lifetime of a circuit, even though NBTI degradation does not fully accumulate until the end of its lifetime. As such, several dynamic, architecture-level approaches [31, 39, 96, 161] have been proposed to mitigate NBTI degradation. Evaluation of architecture-level approaches to mitigate NBTI degradation is typically based on analytical degradation models, like Equation (6.8) [171]:

$$\Delta V_{\text{th}} = A_{NBTI} \cdot \tau_{ox} \cdot \sqrt{C_{ox}(V_{\text{dd}} - V_{\text{th}})} \cdot e^{\frac{V_{\text{dd}} - V_{\text{th}}}{\tau_{ox} E_0} - \frac{E_a}{kT}} \cdot t_{stress}^{0.25} \tag{6.8}$$

where, A_{NBTI} is a constant that depends on the aging rate, τ_{ox} is oxide thickness, C_{ox} is gate capacitance per unit area, E_0, E_a, and k are fitting constants, and, t_{stress} is stress time.

Even though the above equation describes NBTI degradation over time at the device level, its accuracy to evaluate NBTI effect at the architecture-level may be limited, simply because it does not account for scenarios like dynamic voltage scaling, averaging effects across logic paths, and different activity and power management schemes [34].

Recently, many studies have proposed techniques to alleviate the impact of NBTI-induced degradation, from the circuit-level [39, 74, 96, 183, 186] to the architecture-level [3, 31, 78, 158]. At the architecture-level, techniques have been proposed to bias input vectors to mitigate aging [3], enhance throughput at the expense of aging in a multi-core environment [78], monitor and adapt to estimated processor lifetimes [160, 161], perform aging-aware scheduling [158], and apply voltage scaling [171] or power gating [31] to mitigate the effects of aging.

In one of these recent studies, Gupta et al. [34] report that, due to the underlying

physical phenomena that cause NBTI, the degradation is front-loaded by nature. As illustrated in Figure 6.6, this means that the rate of degradation is rapid in the early lifetime and slows down considerably under continued stress.

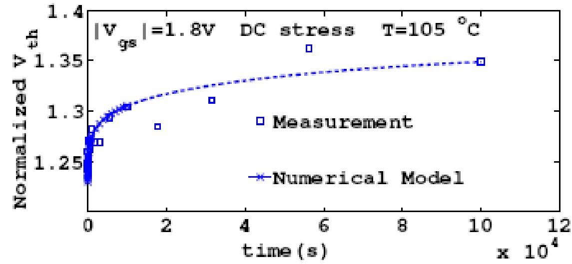


Figure 6.6: An NBTI model [34] vs. measurement data by W. Wang et al. [182].

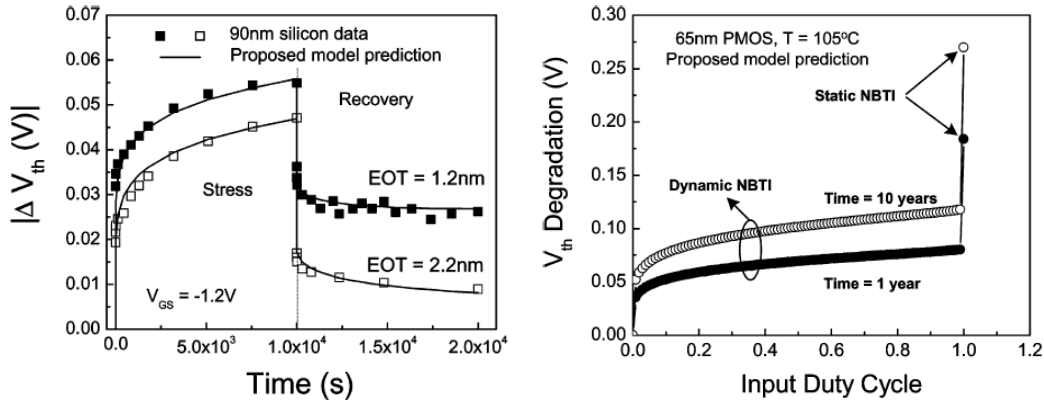
In addition, Gupta et al. [34] (in agreement with this thesis) argue that many of the techniques and evaluations proposed by previous architecture-level publications are not general enough to model the wide range of adaptations and operating scenarios employed by architecture-level NBTI-mitigation techniques. Therefore, the accuracy of these evaluations may be limited. Gupta et al. emphasize that conclusions related to NBTI are strongly dependent on the nature of NBTI degradation. In an effort to mitigate NBTI degradation, Gupta and his research group propose several architecture-level techniques such as dynamic voltage scaling (DVS), lifetime awareness, dynamic instruction scheduling, and power gating—explained in their most recent NBTI work [34].

Consequently, for NBTI analysis, this thesis adopts the method/results introduced by two of the most recent reputable works. The first [184] relies on device-level analytical models and the second [34] utilizes its proposed flexible numerical model for NBTI degradation analysis. We use these two models for our NBTI analysis because they better estimate the

impact of architecture-level techniques on NBTI degradation.

Specifically, in this section, we look at the impact of NBTI on the performance of logic/SRAM circuits under various operating conditions, such as supply voltage, temperature, and node switching activities. We will show that, given a circuit topology and input switching activity, it is possible to efficiently predict the degradation of circuit speed over a long period of time.

The analysis of NBTI is inherently more complicated than that of other traditional reliability issues, such as hot-carrier injection (HCI), explained in section 6.5. NBTI exhibits the unique property of having distinct stress and recovery behavior during a circuit's dynamic operation (Figure 6.7(a)).



(a) V_{th} degradation for dynamic NBTI [184]

(b) Static and dynamic NBTI degradation for different input signal probabilities [184]

Figure 6.7: Impact of V_{th} variation on NBTI.

Depending on the duty cycle and input patterns, over 75% of previous NBTI-induced degradation can be annealed by biasing the pMOS gate at the supply voltage (V_{dd}) [177, 19]. Therefore, the recovery phase and its dependence on node switching activity are critical to the

analysis and design margining for the NBTI-induced degradation. This point is underscored by Figure 6.7(b), which demonstrates that the V_{th} change under dynamic conditions is dramatically different from that in the static mode. Because of the rapid annealing at the beginning stage of the recovery (Figure 6.7(a)), even a small recovery period (i.e., signal probability close to 1) greatly reduces the overall degradation by more than 50% of the static stress. This property is confirmed by silicon data [58, 66] and experimental results. Therefore, an accurate prediction of performance degradation should include not only V_{dd} and T, but also the switching activity of the node. These parameters are not spatially or temporally uniform, but vary significantly from gate to gate and over time due to the uncertainty in circuit topologies and operations. These non-uniformities need to be incorporated into the degradation analysis for both short-term and long-term predictions. Otherwise, a simple static analysis may provide an extremely pessimistic estimation, and consequently, result in drastic over-margining (Figure 6.7(b)). So far, design and tool research are at the early stages in addressing the emerging needs of reliability [184]. The impact of static NBTI on the performance of combinational circuits was analyzed by Paul Bipul C. et al. [133]. Bipul C. et al. demonstrate that by resizing the paths that are most sensitive to NBTI, it is able to mitigate the increase of path delay of the entire circuit. On average, an increase of 8.7% in circuit size is required for 70-nm technology. An algorithm for determining the amount of delay degradation of a circuit due to NBTI is provided by Sanjay V. Kumar et al. [95].

Figure 6.8 illustrates the data flow and structure of the Framework [184] that we have used to estimate the delay degradation due to NBTI. The temporal degradation of circuit performance depends on both technology and design conditions. First, the accurate modeling

of V_{th} degradation at the transistor level is made. For NBTI, predictive transistor models are used to characterize the timing behavior of various basic circuit building gates, such as NAND and NOR gates. An NBTI-aware library is built upon these predictive models. Given a circuit netlist, the new library further supports a timing analysis algorithm that is a simple and efficient way to calculate the circuit performance degradation. By including transistor-level modeling of other reliability mechanisms, such as HCI and NCS, this framework is extendable to analyze other aging effects.

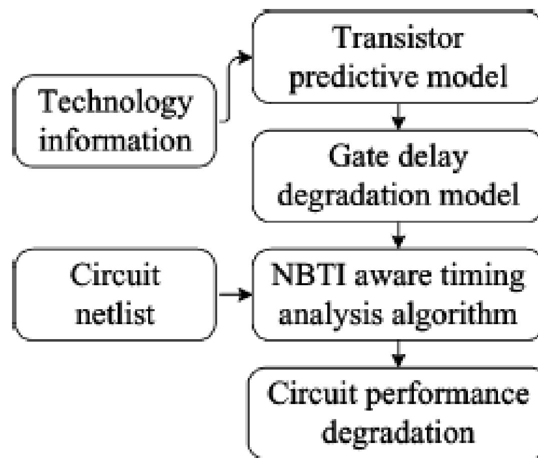


Figure 6.8: NBTI timing analysis framework [184].

Figure 6.9 shows a typical random input sequence within a ten-cycle period—in which there are n “0”s and $(10-n)$ “1”s. An extreme case of such a random sequence is shown in Figure 6.9(b). This input vector has only 1 flip within ten cycles, i.e., is equal to 0.9. This means that the stress time is much longer than the recovery time. Here, the term of $[\alpha/min(\alpha, 1-\alpha)]$ is defined to capture how many cycles are spent in the stress phase. In the case of Figure 6.9(b), this term is equal to 9.

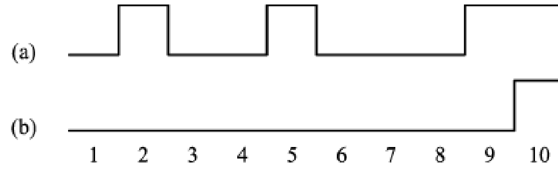


Figure 6.9: Random input sequence. (a) Normal case. (b) Extreme case [184].

Table 6.1: Long term prediction Model of δV_{th} for both periodical and nonperiodical input sequence [184].

$\Delta V_{th,t}$	Periodical	$\Delta V_{th} \leq \left(\frac{\sqrt{K_v^2 \alpha T_{clk}}}{1 - \beta_m^{1/2n}} \right)^{2n}$
	non-Periodical	$\Delta V_{th} \leq \left(\frac{\sqrt{K_v^2 \alpha T_{clk} / \min(\alpha, 1-\alpha)}}{1 - \beta_m^{1/2n}} \right)^{2n}$
β_m	$1 - \frac{2\xi_1 \cdot t_e + \sqrt{\xi_2 \cdot C \cdot (1-\alpha) \cdot T_{clk}}}{2t_{ox} + \sqrt{C \cdot t}}$	

Table 6.1 shows the formulas for the long-term threshold voltage degradation due to NBTI for both periodical and nonperiodical input sequences provided by W. Wang et al. [184]. where T_{clk} is the time period of one stress-recovery cycle, α is the duty cycle (which is the ratio of the time spent in stress to the total time period), βm is the fraction parameter of the recovery, K_v has dependence on electrical field and temperature, n is the time exponent parameter, ξ_1 and ξ_2 are constants, t_e is the effective oxide thickness, C has a temperature dependence as $C = T_0^{-1} exp(-E_a/KT)$ [104], t_{ox} is the oxide thickness, t is the time after which the total number of interface charges are obtained. For more details about the physical meaning of the parameters, refer to S. Bhardwaj‘s work [18].

Since NBTI-induced degradation is relatively insensitive to switching frequency (f) when it is above 100 Hz [18], f is typically fixed at 100 Hz in the experiment (to reduce simulation time) without losing any generality. The simulation results for nodes smaller than 70

nm (i.e. 65-nm and 45-nm) show that there is around an 8% delay degradation in combinational logic circuits after ten years of stress [184]. The simulation results for our 16-nm SRAM circuits show a higher delay degradation of around 10.3% after ten years stress. The expected higher percentage of delay degradation for 16-nm, as compared to those of the larger technology nodes, is due to the thinner oxide thickness, stronger electric field, etc. used in the 16-nm node.

It is imperative to note that despite a considerable amount of research undertaken so far by many researchers (i.e., W. Wang et al.), an accurate and comprehensive understanding of NBTI is not still available to guide reliable design to minimize its impact. Consequently, the results shown in this section should only be considered as some rough estimates, rather than very accurate predictions of circuit aging). Figure 6.10 illustrates the algorithm of circuit timing analysis considering NBTI introduced by W. Wang et al. [184].

To evaluate the timing degradation due to NBTI for each gate in a levelized circuit netlist, three parameters are required: 1) the input pattern for standby mode or the duty cycle of the input for active mode; 2) the slew rate of the input signal; and 3) the gate load capacitance. Given a set of input vectors as the primary inputs of the circuit and assuming that the primary inputs are independent, the duty cycle at the output of any gate can be computed using the duty cycles of its inputs and the logic function implemented by the gate. The degradation of the threshold voltage of a gate in the circuit is then calculated by using the equations of the long-term model (Table 6.1). The slew rate of the first-level gate input signals are defined according to the typical condition of 16-nm design. Once the information is available, including the duty cycle and slew rate for the input signal and output load capacitance, the timing degradation

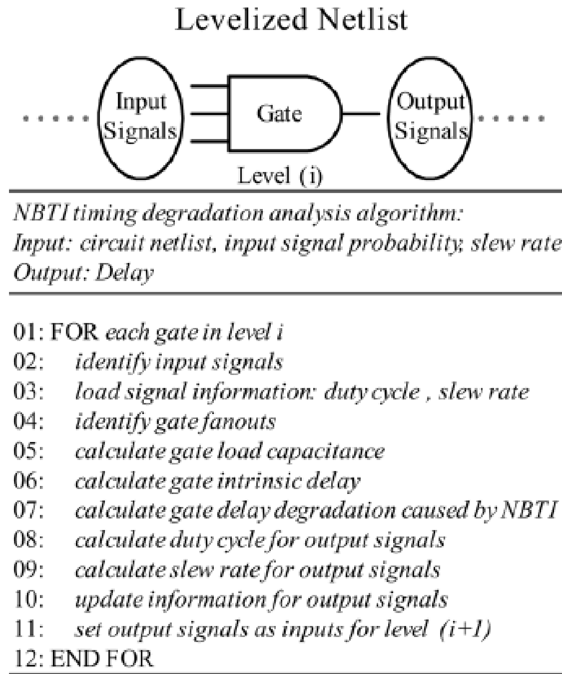


Figure 6.10: Timing degradation analysis algorithm [184].

for the gate under consideration is computed from the NBTI-aware library. By adding this timing degradation to the intrinsic delay of the gate, we obtain the final gate delay. At the same time, the library model uses the slew rate of the input signals, gate load capacitance, and gate threshold voltage degradation to calculate the slew rate of the output signal. Signal duty cycle and slew rate are propagated from level to level, and the earlier timing analysis procedure is repeated until the timing degradation of the final level is calculated.

The following three sub-sections describe the results and key insights obtained by applying the NBTI timing analysis of W. Wang et al. [184] on our 16-nm SRAM circuits. We set f = 100Hz for the analysis in these sub-sections.

6.4.1 Supply Voltage and Temperature Dependence

NBTI has strong dependence on V_{dd} and T [178, 19]. Here, V_{dd} refers to the operating supply voltage for a given circuit. The nominal V_{dd} is assumed to be 0.7V and the nominal T is 80 C. The data for the 16-nm V_{dd} and T profiles are extrapolated from data for an industrial 65-nm design provided by Wang et al. [184]. Based on the extrapolated data, the variations of V_{dd} and T for the whole chip are assumed to be within 10% for NBTI analysis. For the purpose of circuit timing analysis, we follow Cao's method [184] and select five representative operating conditions with different combinations of V_{dd} and T: high V_{dd} and high T (HH), low V_{dd} and low T (LL), high V_{dd} and low T (HL), low V_{dd} and high T (LH), and normal V_{dd} and normal T (NN). In order to analyze the temperature dependence in a wider range, we also include one more condition: low V_{dd} and room temperature (LL'). Using the formula, algorithm, and procedure outlined in this section, we obtain the delay degradation for different SRAM circuits after one year, five years, and ten years stress—as illustrated in Table 6.2.

Table 6.2: Simulation results for two 16-nm SRAM circuits: *arcN* (non-optimum, $\frac{4:64:256}{1:1:1}$) and *arcO* (optimum, $\frac{64:64:16}{1:1:1}$)

Circuit	1 Year (%)						5 Years (%)						10 Years (%)					
	NN	HH	LH	LL	HL	LL'	NN	HH	LH	LL	HL	LL'	NN	HH	LH	LL	HL	LL'
(arcN)	9.9	11.8	11.5	9.5	9.2	6.1	13.9	15.4	16.3	13.1	12.6	8.1	16.2	17.8	19.2	14.1	14.7	10.5
(arcO)	6.7	7.3	7.7	6.5	6.3	4.4	8.8	9.6	10.1	8.4	8.2	5.7	9.9	10.8	11.2	9.6	9.3	6.4

From Table 6.2, we conclude the following three important observations for dynamic circuit operation:

1. Temperature has a bigger impact on the degradation of circuit performance than the operating supply voltage. For instance, after ten years stress, the delay degradation of circuit

arcN is about 19.2% under the LH condition, while it is about 14.1% under the LL condition. The degradation difference caused by temperature is about 5%. If we further reduce T to room temperature, the delay degradation can be reduced to about 10.5%. Therefore, lowering the temperature is a very effective approach to minimize NBTI.

2. Within the allowed 10% voltage variation, tuning the operating V_{dd} does not show any advantage in reducing NBTI. For example, the delay degradation of circuit *arcO* is about 7.7% under the LH condition, while it is 7.3% under the HH condition. The degradation difference caused by voltage is only 0.4%.

3. Although lower operating V_{dd} is generally preferred to reduce the amount of circuit aging, it does not hold true for scaled CMOS design, as observed in our simulation results. On the contrary, lower operating voltage may lead to more circuit timing degradation for the 16-nm technology node, as shown in Figure 6.11. Given the stress time, there exists an optimum operating V_{dd} that achieves the minimum amount of circuit delay degradation. When V_{dd} is lower than that value, circuit performance becomes increasingly sensitive to changes in V_{th}, and thus, the degradation rate climbs even though the absolute increase of V_{th} is smaller than that at higher V_{dd}. On the other hand, when V_{dd} is higher than that value, the amount of V_{th} increases exponentially, dominating the performance degradation. The exact value of the optimum operating V_{dd} also depends on the technology node and the circuit structure.

In summary, during dynamic operation, NBTI-induced degradation is relatively insensitive to supply voltage, but strongly dependent on temperature. In addition, there is an

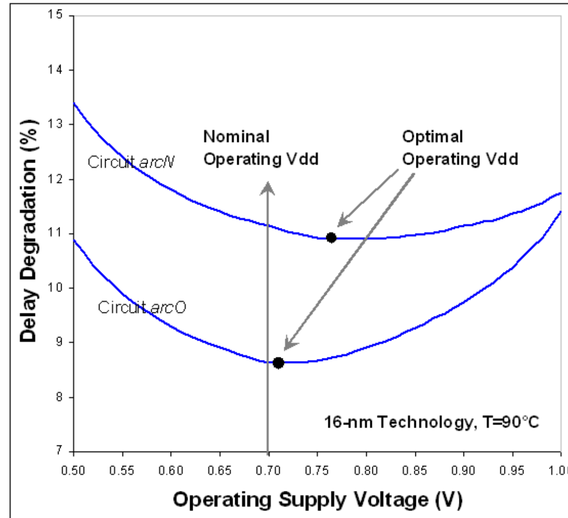


Figure 6.11: Optimal V_{dd} for minimum degradation of circuit performance for two different 16-nm SRAM architectures: optimal ($\frac{64:64:16}{1:1:1}$) and non-optimal ($\frac{4:64:256}{1:1:1}$).

optimum supply voltage that leads to the minimum circuit performance degradation; the circuit degradation rate actually goes up if the supply voltage is lower than that optimum value. Since our simulation results agree with those of [184], we have confidence that the NBTI analysis presented in this section is valid.

6.4.2 Input Control in Static and Dynamic Operation

In addition to the dependence on V_{dd} and T, NBTI has an optimum gate voltage, as well. For a pMOS, a gate bias at V_{dd} helps the recovery, while a gate bias at "0" stresses the transistor. A longer time spent in recovery (i.e., lower duty cycle) corresponds to smaller changes in V_{th} for the transistor. Because of this mechanism, NBTI is strongly affected by node activity. In standby mode, this implies a dependence on input patterns; during the dynamic operation, the duty cycle further impacts the relative time between stress and recovery [184].

1. ***Input Pattern Dependence:*** For a circuit containing n inputs, each input signal can be either set to "1" or "0" during the standby mode. Thus, the circuit can have at most 2^n possible input patterns. Since NBTI has a strong dependence on the input pattern of the circuit, different input patterns will result in significantly different delay degradations. An input vector that results in the *least* delay degradation of the circuit is referred to as the *best standby mode*. Similarly, an input vector that results in the *most* delay degradation is referred to as the *worst standby mode*. Similar to Wang et al. [184], we estimate the *best* and the *worst standby mode* by sampling the circuit with 500 different input vectors. By biasing several selected SRAM circuits under the *worst* and the *best standby modes*, we compare their delay degradations for one, five, and ten year periods and record the results. Based on the results, we see that the delay degradation caused by NBTI can be greatly reduced by applying the optimal input pattern to the entire circuit in the standby mode. A typical example is circuit $\left\{\frac{8:64:128}{1:1:1}\right\}$. After ten years, the delay degradation for the worst standby mode is about 46%, while under the best standby mode it is about 11%. The delay degradation can change by more than a factor of 4 for different input patterns. Like NBTI, the leakage current of a circuit also has a strong dependence on the input pattern. Therefore, if the application in which the SRAM is used allows a set of pre-selected input patterns in the standby mode, both the temporal degradation caused by NBTI and the circuit leakage can be minimized. Again, this result is validated by its similarity to the results of Wang et al. [184].

2. ***Duty Cycle Dependence:*** For a circuit operating in the dynamic mode, the probability

that each input can take a value of “1” or “0” can be any value between 0 and 1. For a given circuit with n inputs, α_i, $i \, \varepsilon \{1,...,n\}$ is the duty cycle of input i. Like Wang et al. [184], we define one combination of $\{\alpha_1, \alpha_2, ..., \alpha_n\}$ as one α set. Since for an n-input circuit, the number of distinct α sets can be infinite, we choose five typical values in order to analyze the impact of different α sets on the circuit performance: 0.1, 0.3, 0.5, 0.7, and 0.9 for each α_i. That means in the α sets, all α are set to either 0.1, 0.3, 0.5, 0.7, or 0.9.

To observe how the duty cycle affects the delay degradation of circuits over time, we apply the formula/methods outlined earlier in Table 6.1 and Figure 6.10. We use three different α sets on two of our selected SRAM architectures (where I stands for $\left\{\frac{4:64:256}{1:1:1}\right\}$ and II stands for $\left\{\frac{64:64:16}{1:1:1}\right\}$). We observe that, within the same architecture, different α sets can result in very different timing degradation. For example, after one year of stress, the delay degradation of circuit I (the bottom curves, Figure 6.12) with an input duty cycle of α set3 is nearly 2× larger than that with α set1. In addition, the difference in delay degradation (Δ) increases with time, i.e., Δ_2 is much larger than Δ_1. As mentioned previously, NBTI is clearly related to the gate bias due to its exponential dependence on the electrical field. Therefore, for a circuit operating in the dynamic mode, NBTI-induced degradation can be reduced by adjusting the input signal α such that it stays in the recovery state longer.

Figure 6.12 illustrates that the delay degradation profile of the circuit after ten years has a much wider spread than the degradation after one year. Meaning that, with increasing

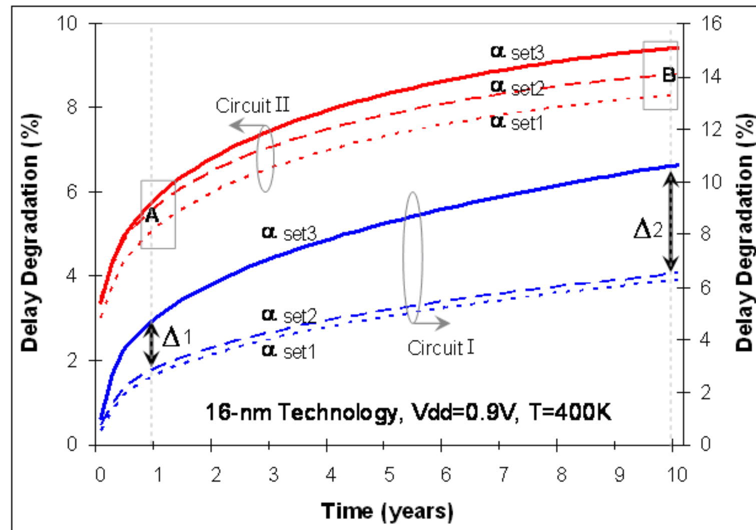


Figure 6.12: Delay degradation over time for various duty cycle sets of two sample circuits (circuit I and circuit II).

time, different α sets tend to generate diversified effects on the circuit degradation. In other words, several α sets might result in a similar circuit delay degradation in a short time period. However, in the long term, they can result in very different degradations. Furthermore, experiments [184] show that using a higher number of input α sets tends to generate a larger timing degradation and the path delay distribution becomes wider in the long run. This is because different input αs result in very different ΔV_{th} (Figure 6.7(b)), which correspondingly leads to wide distribution of circuit path timing. Therefore, modulating node activities will be a very useful design tool to mitigate NBTI for dynamic operation. The use of this tool, however, has its own limitations and comes with some disadvantages if used for critical paths in SRAM circuits. For example, the reduction of duty cycles on such signals as WRITE, WL (wordline pulse), or CLS (pulse for senseamp) can reduce the stability and robustness of SRAM.

In summary, circuit performance degradation due to NBTI is highly sensitive to input vectors. The difference in delay degradation could be up to 5× for various static and dynamic operations.

6.4.3 Impact of NBTI on Process/Design)

As discussed earlier, NBTI originates from a transistor-level phenomenon, and it can interact with many other process and circuit parameters. Based on the earlier simulation framework, we further examine these interactions with process variability and operation uncertainty.

1. ***Interaction between NBTI and Process Variability:*** Process variations, such as random dopant fluctuations, add great uncertainty to scaled-down circuit design. Since NBTI-induced transistor and circuit performance degradations are highly sensitive to process parameters and operation conditions (including V_{th}, temperature, and switching activity), circuit aging strongly interacts with static process variations. Under NBTI, a pMOS device with lower V_{th} degrades much faster than with a higher value of V_{th}, and thus, its V_{th} increases more after the degradation. As a result, the difference in V_{th} among different transistors becomes smaller after some period of stress. Figure 6.13 shows the frequency change of an 11-stage ring oscillator (RO) over time [184]. At time 0, the difference between low V_{th} and high V_{th} is 60 mV, which results in 6.2% variation in frequency. At ten years, the variation reduces to 1.6%. As time increases, the frequency difference due to process variations decreases. From Figure 6.13, we also observe that the frequency degradation caused by NBTI after ten years is 10.5%, which is more than the difference caused by process variations at t = 0 [184]. For robust circuit design, this

information implies that both temporal changes under NBTI and static process variations must be considered.

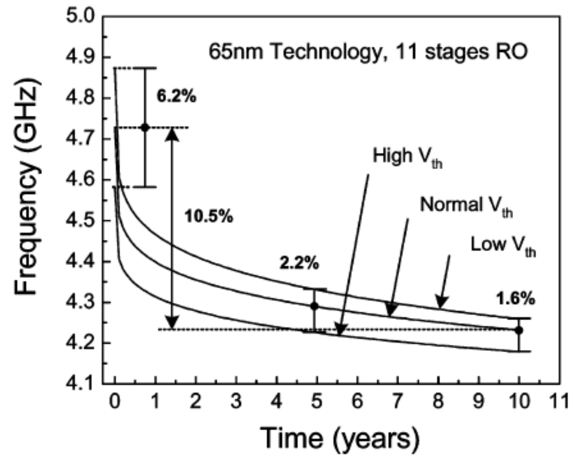
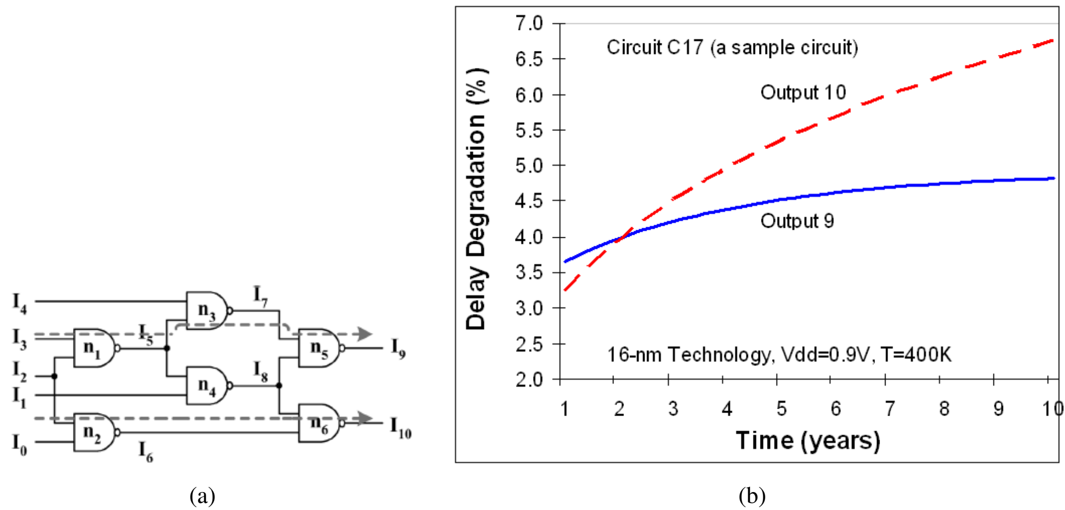


Figure 6.13: Frequency degradation of an 11-stage ring oscillator (RO) under both process variation and NBTI effect [184].

2. ***Impact of NBTI on Path Reordering:*** The reordering of the critical path is one of the most important impacts of NBTI on circuit timing. For traditional static timing analysis, the paths of a multiple-output circuit have a fixed timing order. However, with NBTI, the original critical path may become non-critical and vice versa, since the gate delay degradation is strongly influenced by the input duty cycle and sequence, which are uncertain in real operation. Such path reordering is likely to happen under NBTI: originally, at time t = 0, NBTI is not in effect and the critical path has a larger delay than the non-critical path. We assume that the non-critical path is more sensitive to NBTI under α set1, whereas the critical path is not sensitive to α set1; on the other hand, the non-critical path is not sensitive to α set2 and the critical path is highly sensitive to α set2. With increasing time and with the input signal switching from α set2 to α set1, the critical path

and the non-critical path may experience different amounts of degradation and eventually switch their roles. To illustrate this effect, we simulate circuit C17 (a benchmark circuit suggested by Wang et al. [184]) for the 16-nm node.



(a) (b)

Figure 6.14: Example circuit to demonstrate the critical path changing with time. (a) C17 benchmark circuit. (b) Timing degradation versus time.

Figure 6.14(a) shows its circuit netlist and Figure 6.14(b) is the delay degradation over time. At time t = 0, the outputs 9 and 10 have the same delay. We then bias the circuit to α set1 and that results in output 9 degrading more than output 10. After some time, we change the input to α set2. Eventually, the arrival time of output 10 surpasses that of output 9. In traditional design optimization, one basic objective is to identify the critical path of the circuit, size up the gates in the critical path for performance speedup, and size down the gates in the non-critical path for power and area reduction [184]. Due to NBTI-induced path reordering, NBTI-aware timing analysis and optimization will be more complicated and will require innovative solutions since a a large set of potential critical paths need to be optimized.

In summary, NBTI leads to a temporal reduction of the threshold voltage in MOSFETs and temporal frequency degradation in circuits. In addition, NBTI uncertainty can swap the orders of critical and non-critical paths over time.

6.5 Hot-Carrier Injection (HCI)

Hot carrier injection (HCI) is a phenomenon in solid-state electronic devices where an electron or a “hole” gains sufficient kinetic energy to overcome the potential barrier (also called “quantum tunneling”) necessary to break an interface state. The term “hot” refers to the effective temperature used to model carrier density, not to the overall temperature of the device. That is, high temperatures caused by the effect are unrelated to the phrase “*hot electron effect*.” The term “*hot carrier injection*” usually refers to the effect in MOSFETs, where a carrier is injected from the conducting channel in the silicon substrate to the gate dielectric, which is usually is made of silicon dioxide (SiO2). To become “hot” and enter the conduction band of SiO2, an electron must gain a kinetic energy of 3.3 eV. For holes, the valence band offset in this case dictates they must have a kinetic energy of 4.6 eV. Since the charge carriers can become trapped in the gate dielectric of a MOS transistor, the switching characteristics (V_{th}) of the transistor can be permanently changed. Hot-carrier injection is one of the mechanisms that adversely affects the reliability of semiconductors in solid-state devices [80].

The *hot electron effect* occurs in semiconductor devices when electrons are excited to energy levels higher than those associated with the semiconductor‘s conduction band. Instead of recombining with a hole or being conducted through the material to a collector, these *hot*

electrons can tunnel out of the semiconductor material. Because *hot electrons* generally lose excess energy via phonons, a common manifestation of the *hot electron effect* is an increase in the heat of the semiconductor device [170]. In some semiconductor devices, this represents an inefficiency as energy is lost as heat. For instance, some solar cells rely on the photovoltaic properties of semiconductors to convert light to electricity. In such cells, the *hot electron effect* is the reason that a portion of the light energy is lost to heat rather than converted to electricity [170]. Another consequence is increased leakage current.

In MOSFETs, *hot electrons* have sufficient energy to tunnel through the thin oxide gate and show up as gate current, or as substrate leakage current. The *hot electrons* may come from the channel region or from the drain, for instance, and travel into the gate or the substrate. For example, in a MOSFET, when a gate is positive and the switch is on, the device is designed so that electrons will flow through the conductive channel to the drain. These *hot electrons* do not contribute to the amount of current flowing through the channel as intended and, instead, are a leakage current.

Attempts to correct or compensate for the *hot electron effect* in a MOSFET may involve placing a reverse-bias diode at the gate terminal or other manipulations such as lightly doped drains or double-doped drains.

When electrons are accelerated in the channel, they gain energy along the mean free path. This energy is lost in two different ways:

1. The carrier hits an atom in the substrate. This collision creates a cold carrier and an additional electron-hole pair. In the case of nMOS transistors, additional electrons are

collected by the channel and additional holes are evacuated by the substrate.

2. The carrier hits a Si-H bond and breaks the bond. An interface state is created and the H atom is released into the substrate.

The probability of intercepting either an atom or a Si-H bond is random, and the average energy involved in each process is the same in both cases. This is the reason why the substrate current is monitored during HCI stress. A high substrate current means a large number of electron-hole pairs have been created, pointing to the existence of an efficient Si-H bond breaking mechanism. When interface states are created, the threshold voltage is modified and the sub-threshold slope is degraded. This leads to lower current, and reduces the operating frequency of the integrated circuit. As pointed out in Section 6.4, by including transistor-level modeling of HCI to the framework used in section 6.4, the analysis of aging and performance can be extended to include the impact of HCI. There are a number of models to describe the hot-electron effect [52]. Due to the similarity between HCI effects and NBTI, this thesis leaves such inclusion to future work.

6.6 Single Electron Tunneling (SET)

Single Electron Tunneling (SET) technology uses a single electron (or a few) to implement various analog and digital applications, such as memory. However, many of the existing SET-based memory cells proposed so far may not work properly if there are random noise sources present, such as capacitance variations, supply voltage variations and especially background charges on the circuit nodes/islands [24].

If and when SET becomes sufficiently reliable for commercial use, it will replace today's SRAM, DRAM, etc. with a new type of memory cell due to its numerous advantages. For example, SET is characterized by much smaller feature sizes, which allows smaller, faster, and more energy-efficient memory cells. However, as of today, nobody has shown any solid evidence that SET will be able to replace the existing memory technologies in the future. We included this section only to make memory designers aware of this research—which has not yet been commercially successful!

Chapter 7

Design Considerations and Analysis, Power

7.1 Impact of Temperature on Delay, Power, and Performance

An increase in temperature impacts the performance of SRAM. Both the delay and the power suffer from an increase in the circuit temperature mainly due to the adverse impact of temperature on the drain current and interconnect resistance. Therefore, in our analysis of the temperature-dependency of the delay, power, and performance of SRAM, we consider both the change in the drain current of the transistors on the critical path and the change in the wire resistance of the bitlines and wordlines.

First, the analysis of the drain current and its dependency on temperature is the basis for the delay propagation (t_p) and leakage current (I_{leak}), both of which relate to the temperature-dependent parameters used in the drain current. Referring to the drain current equations below (Equations (7.1) and (7.2)), the temperature affects certain variables such as the mobility and threshold voltage which determine I_d (drain current), I_{dsat} (drain current in saturation), and Req

(equivalent resistance) of the transistors on the critical path. The two equations correspond to the saturation and triode modes, respectively:

$$Id = \mu Cox \frac{W}{L}(Vgs - V_{\text{th}})^2 \qquad (Saturation) \tag{7.1}$$

$$Id = \mu Cox \frac{W}{L}(Vgs - V_{\text{th}})Vmin - \frac{Vmin^2}{2} \qquad (Triode) \tag{7.2}$$

where μ is the charge-carrier effective mobility, Cox (which is equal to εox/tox) is the gate oxide capacitance per unit area, W is the gate width, L is the gate length, Vgs is the potential difference between the gate and source of a transistor, V_{th} is the threshold voltage, and Vmin represents the potential difference between the drain and source of a transistor.

During the *access* operation (and *read* operation), the access transistor (A_L in Figure 7.1) is in saturation mode and the pull-down transistor (N_L in Figure 7.1) is in triode mode. A similar principal applies to the *write* operation, except that instead of the left access transistor A_L operating in the saturation mode, the right access transistor A_R operates in the triode mode while the pull-up transistor P_R operates in the saturation mode.

An increase in temperature decreases the mobility, μ, as shown in the following equation:

$$\mu(T) = \mu_0 (T/T_0)^{\alpha_\mu} \tag{7.3}$$

Typical electron mobility for Si at room temperature (300 K) is 1400 $cm^2/(V \cdot s)$ and

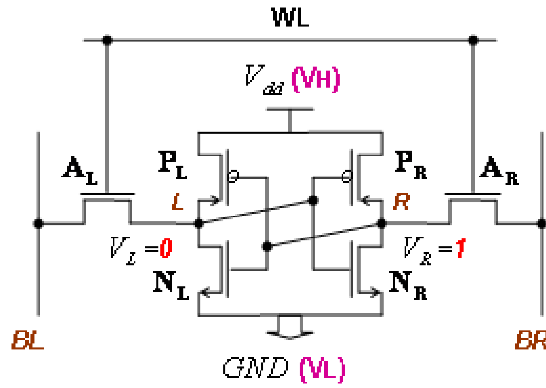


Figure 7.1: 6 transistor (6T) storage cell (repeated for convenience).

the hole mobility is around 450 $cm^2/(V \cdot s)$.

Similarly, an increase in temperature leads to a decrease in the threshold voltage, as shown in the following equation:

$$V_{\text{th}}(T) = V_{\text{th0}} + \alpha_{\text{Vth}}(T - T_0) \tag{7.4}$$

Temperature dependency of mobility, threshold voltage and resistance along with their typical values for the parameters used in Equations (7.3) and (7.4) are summarized/shown in Table 7.1.

According to Equations (7.3) and (7.4), both the mobility (μ) and the threshold voltage (V_{th}) decrease with an increase in temperature. However, the decrease in μ is slightly larger than the decrease in V_{th}, comparatively. Looking back at Equations (7.1) and (7.2), we observe that the impact of a temperature increase on the drain current will not be dramatic, simply because the changes in $V_{\text{th}}(T)$ and $\mu(T)$ are approximately equal and opposite in sign. The authors of VARIUS [169] express the partial cancellation of μ and V_{th} temperature dependency

Table 7.1: Temperature dependency of mobility, threshold voltage and resistance [191].

$\mu(T) = \mu_0(T/T_0)^{\alpha_\mu}$ $V_{\text{th}}(T) = V_{\text{th0}} + \alpha_{\text{Vth}}(T - T_0)$ $R(T) = R_0[1 + \alpha_{\text{R}}(T - T_0)]$ where, T is the temperature, T_0 is the nominal temperature, μ_0 is the mobility at T_0, V_{th0} is the threshold voltage at T_0, R_0 is the resistance at T_0; α_μ, α_{Vth}, and α_R are empirical terms named the mobility temperature exponent, threshold votage temperature coefficient, and resistance temperature coefficient, respectively, where α_μ = -2, -1 mV/ °C$\leq \alpha_{\text{Vth}} \leq$-4 mV/°C, and α_R in Cu is 0.004.

by illustrating the relation between these two parameters in the toggling frequency equation below:

$$T_g \propto \frac{L_{\text{eff}} V}{\mu(V - V_{\text{th}})^{\alpha}} \tag{7.5}$$

where α is typically 1.3 and μ is the mobility of carriers ($\mu(T)\ propto\, T^{-1.5}$). As V_{th} decreases, $V - V_{\text{th}}$ increases and the gate becomes faster. As T increases, $V - V_{\text{th}}(T)$ increases, but $\mu(T)$ decreases [169]. The second factor dominates and, with higher T, the gate becomes slower, though not dramatically.

Equations (7.6), (7.7), and (7.8) show in more detail how V_{th} is decreased by increase in temperature through such parameters as γ (body effect) and ϕ_F (surface potential).

$$V_{\text{th}} = V_{\text{th0}} + \gamma(\sqrt{|(-2)\phi_F + V_{SB}|} - \sqrt{|2\phi_F|}) \tag{7.6}$$

$$\gamma = (t_{ox}/\varepsilon_{ox})\sqrt{2q\varepsilon_{si}N_A} \tag{7.7}$$

$$\phi_F = (kT/q)ln(N_A/N_i) \tag{7.8}$$

where

t_{ox} is oxide thickness,

ε_{ox} is oxide permittivity, $\varepsilon_{ox} = \varepsilon_{Si} \times \varepsilon_0$,

ε_{Si} is the relative dielectric constant of silicon,

ε_0 is the permittivity of free space,

q is the charge of an electron, 1.602×10^{-19} C,

K is Boltzmann‘s constant, 8.6173324×10^{-5} eV KE-1,

T is Temperature in Kelvin, 300K,

N_A is the doping concentration for substrate, and

N_i is the intrinsic doping concentration for the substrate.

Having looked at the impact of temperature on the transistor drain current above, we will now discuss the propagation delay of a gate (i.e., an inverter). The previously described temperature effects of the drain current impact the delay of a gate and therefore the delay and performance of SRAM. Equation (7.9) shows the overall propagation delay, t_p, of an inverter [141].

$$t_p = \frac{t_{pHL} + t_{pLH}}{2} = 0.69C_L(\frac{R_{eqn} + R_{eqp}}{2}) \tag{7.9}$$

This analytical equation assumes that the equivalent load-capacitance, C_L, is identical for both the high-to-low, t_{pHL}, and low-to-high, t_{pLH} transitions, with R_{eqn} and R_{eqp} representing the equivalent *on-resistance* of the NMOS and PMOS, respectively. Typically, the on-resistance of NMOS and PMOS are set to be approximately equal (through transistor sizing) so that they have identical propagation delays for both rising and falling inputs.

C_L in Equation (7.9) represents the total load capacitance, which is composed of input, diffusion and gate capacitances of the NMOS and PMOS transistors of the inverter [141]. C_L increases as the temperature increases mainly due to the junction capacitance (C_j), affecting the diffusion capacitances of the NMOS and PMOS transistors. C_L also increases due to K_{eqn} and/or K_{eqp}, but only slightly. (K_{eqn} and/or K_{eqp} are multiplication factors for NMOS and PMOS, respectively, and relate the linearized capacitor to the value of the junction capacitance under the zero-bias condition.) Equations (7.10) through (7.15) show the temperature-dependency of C_L. (The equations for other components of C_L, namely, input and gate capacitances that have little to no dependency on temperature are not shown to avoid cluttering.)

$$Cdb1 = K_{eqn} \cdot ADn \cdot C_j + K_{eqswn} \cdot PDn \cdot CJSW \tag{7.10}$$

$$Cdb2 = K_{eqp} \cdot ADp \cdot C_j + K_{eqswp} \cdot PDp \cdot CJSW \tag{7.11}$$

$$C_{eq} = K_{eq} \cdot C_{j0} \tag{7.12}$$

$$C_j = \frac{C_{j0}}{\sqrt{1 - \frac{V_D}{\phi_0}}} \tag{7.13}$$

$$\phi_0 = \phi_T ln[\frac{NA \times ND}{n_i^2}] \tag{7.14}$$

$$\phi_T = \frac{KT}{q} = 26mV\ at\ 300K \tag{7.15}$$

where,

$Cdb1$ and $Cdb2$ are drain-to-bulk diffusion capacitances,

ADn and ADp are areas of the drain and PDn and PDp are perimeters of the drain of NMOS and PMOS transistors, respectively,

K_{eq} is the coefficient of junction capacitance under zero-bias,

CJSW is the side-wall junction capacitance,

C_{j0} is the depletion-layer capacitance per unit area under zero-bias conditions,

ϕ_0 is the voltage across the junction called the built-in potential,

ϕ_T is the thermal voltage,

NA and ND are the acceptor and donor concentrations, respectively, and

n_i is a quantity representing the intrinsic carrier concentration in a pure sample of the semiconductor and equals approximately $1.5 \times 10^{10} cm^{-3}$ at 300 K for silicon.

R_{eq} in Equation (7.9) is related to the saturated drain current, I_{dsat}, which in turn, is related to μ and V_{th} (two temperature-dependent components of the drain current) through the

following equations [141]:

$$R_{eq} = \frac{1}{\frac{V_{dd}}{2}} \int_{\frac{V_{dd}}{2}}^{V_{dd}} \frac{V}{I_{DSAT}(1+\lambda V)} dV = \frac{3}{4} \frac{V_{dd}}{I_{DSAT}} (1 - \frac{7}{9} \lambda V_{dd}) \tag{7.16}$$

$$with \;\; I_{DSAT} = \mu C_{ox} \frac{W}{L} \left[(V_{dd} - V_{th}) V_{DSAT} - \frac{V_{dsat}^2}{2} \right] \tag{7.17}$$

where λ is an empirical parameter called the channel-length modulation. In general, λ is proportional to the inverse of the channel length. In shorter transistors (such as those used in 16-nm technology), the drain-junction depletion region presents a larger fraction of the channel, and the channel-modulation effect is more pronounced. V_{DSAT} is the saturation drain voltage.

In addition to its impact on the drain current, an increase in temperature results in an increase in the resistivity (R) and therefore the delay, t, of the interconnects (also called Elmore delay). Equations (7.18), (7.19), and (7.20) show the temperature dependency of the wire delay:

$$R(T) = R_0[1 + \alpha_R (T - T_0)] \tag{7.18}$$

$$t = ln(2)\tau = 0.69\tau = 0.69 \times R \times C \tag{7.19}$$

$$R = \rho \frac{l}{A} \tag{7.20}$$

where,

R is the resistance of a conductor of uniform cross section length, measured in meters (m),

A is the cross-sectional area of the conductor measured in square meters (m^2), as shown in Figure 7.2,

ρ (rho) is the electrical resistivity (also called specific electrical resistance) of the material, measured in ohm-meters ($\Omega \cdot m$),

τ is the time constant of the interconnect, and

α_R is the resistance temperature coefficient of the interconnect. As shown in Table 7.1, the typical value of α_R in copper (Cu) is 0.004.

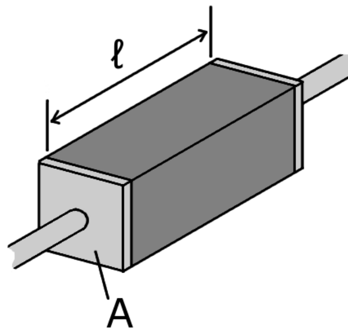


Figure 7.2: A piece of resistive material with electrical contacts on both ends [101].

By applying the equations presented above to 16-nm technology, we can observe the impact of temperature on mobility (μ), threshold voltage (V_{th}), and wire resistance (R) (Figure 7.3) and the impact of temperature on I_{dsat} and I_{triode} (Figure 7.4).

Figure 7.3 illustrates that both mobility (μ) and threshold voltage (V_{th}) decline as the temperature rises from 27°C to 125°C. From the plots, we can see that while μ decreases non-linearly with the increase in temperature, V_{th} decreases linearly and more moderately as compared to μ. Figure 7.3 helps visualize why the impact of temperature on the drain current (Equations (7.1) and (7.2)) is not dramatic: the sign of V_{th} is negative in those equations, therefore, it partially cancels the changes in μ.

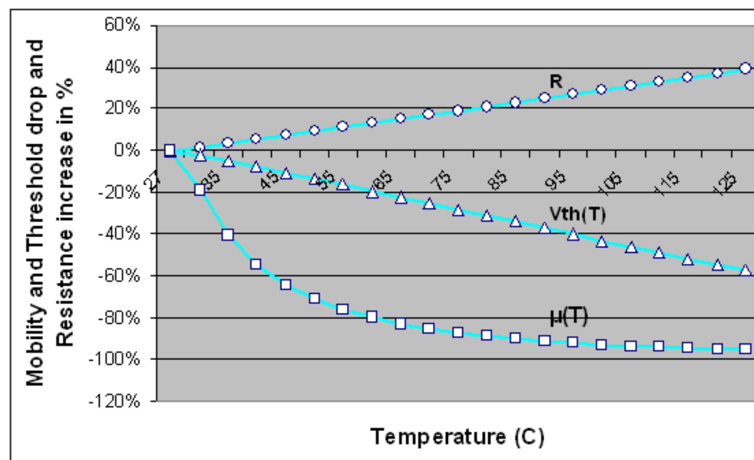


Figure 7.3: NMOS Mobility & Threshold, and wire Resistance change vs. Temperature.

The impact of the changes in μ and V_{th} due to temperature (shown in Figure 7.3) on the drain current, I_{dsat} and/or I_{triode}, is shown in Figure 7.4. For the temperature range of 27°C to 125°C, the average I_d drop is less than 1% per $°C$. It is interesting to note that this percentage varies for different temperature ranges. For example, the I_d drop is more than 2% per $°C$ for the more limited temperature range of 27 to 55°C. This is simply due to the non-linear nature of the drain currents.

Figures 7.3 and 7.4 also show the behavior of the resistance (R), and the corresponding behavior of the wire delay(determined by Equations (7.18), (7.19), and (7.20). The wire

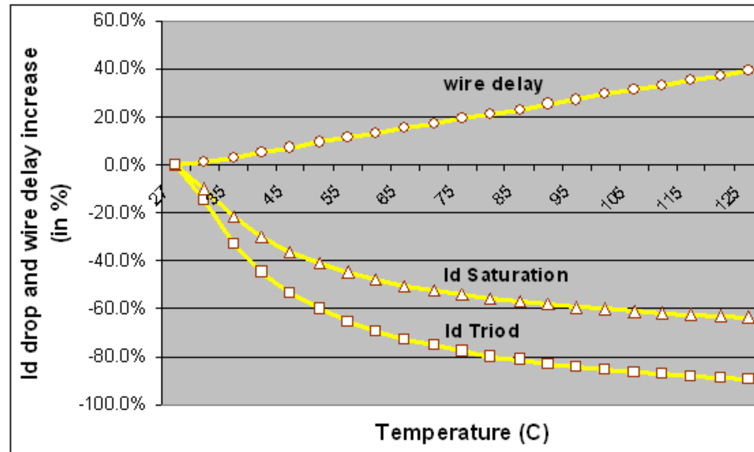


Figure 7.4: Drain Current and Wire Delay vs. Temperature.

delay has a linear relation with the wire resistivity and increases with temperature at an approximate rate of 0.4% per degree centigrade. This rate could vary if the material used for the interconnect is different from the typical copper or M1, M2. Studies show that the increase in temperature of the interconnects used in SRAM chips is about 6.8°C, and this translates into approximately a 2.72% increase in the interconnect delay [191]. This means the impact of temperature on the delay of typical interconnects used in advanced SRAM chips today is not significant, especially as compared to the impact of temperature on leakage current [169] (that will be discussed shortly). In Chapter 10, we will see how the combined drain current decrease and wire delay increase (due to temperature increase) impact the access-time, leakage current, and performance of 16-nm 64Kb SRAM.

It is instructive to note that designers can choose from several possible techniques to reduce the impact of temperature on SRAM performance. One technique is designing with a V_{th0} computed at a higher temperature (e.g. 85°C rather than 27°C, room temperature). That way, the delay will be insensitive to T = 85°C. Another technique is using a well-chosen seg-

ment length for wordlines and bitlines. Choosing an appropriate length for each of the bitline segments can reduce the leakage current and therefore reduce the impact of temperature on total leakage. Similarly, choosing an appropriate length for each of the wordline segments reduces the resistivity of the interconnect delay and therefore reduces the temperature impact on the wordline delay. Using larger logical effort buffering also helps in the reduction of temperature effects at the expense of some extra area and total power. Other techniques (such as throttling, TRC, etc.) attempt to reduce the temperature increase in the SRAM/cache/chip and are discussed in the next section.

7.2 Temperature and Voltage Variation

Systematic and random variations in process, supply voltage, and temperature (P, V, T) are posing a major challenge to the future of high performance micro-processor design that incorporate SRAMs as part of their cache component. Technology scaling beyond 90-nm has caused higher levels of device parameter variation, which have changed the design problem from deterministic to probabilistic over the last few years [153, 77]. This variation will be even higher going towards the 16-nm or 8-nm technology nodes. The demand for lower power requires supply voltage scaling, and thus, voltage variations are becoming a significant part of the overall challenge [27]. Finally, the quest for increasing operating frequency has resulted in significantly high junction temperature and within-die temperature variation. We have briefly discussed the impact of P (Process) variations on circuits in section 6.1, and we will discuss the impact of V and T variations on circuits and micro-architecture in this section. Our discussion will include presenting some possible solutions to reduce or tolerate the parameter variations (e.g., temperature, voltage) in high frequency circuit designs, suggested by some well-received studies.

7.2.1 Supply Voltage Variation

Variations in switching activity and diversity in the type of logic result in uneven power dissipation across the die. This variation results in uneven supply voltage distribution and temperature hot spots, causing transistor sub-threshold leakage variation. The supply voltage (V_{cc}) will continue to scale modestly by 15%, not by the historic 30% per generation, due to 1)

difficulties in scaling threshold voltage (V_{th}) and 2) to meet the transistor performance goals. Maximum V_{cc} is specified as a process reliability limit and minimum V_{cc} is required for the target performance [27]. V_{cc} variation inside the max-min window is shown in Figure 7.5. This figure shows a drop in V_{cc}, which degrades the performance. Packaging and platform technologies do not follow the scaling trends of CMOS processes. Therefore, power delivery impedance does not scale with V_{cc} and ΔV_{cc} has become a significant percentage of V_{cc}.

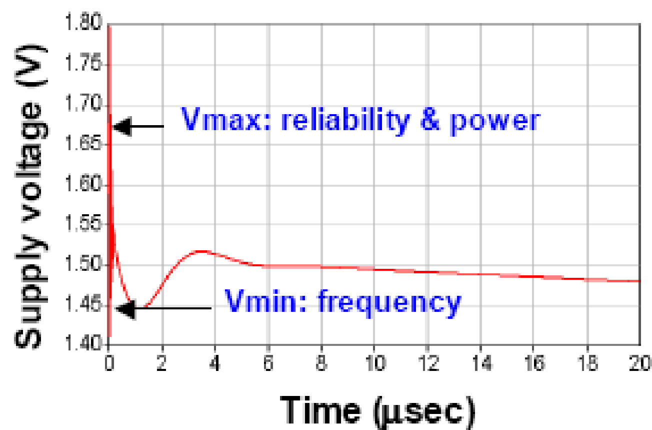


Figure 7.5: Supply voltage variation [27].

7.2.2 Temperature Variation

Figure 7.6 shows the thermal image of a leading micro-processor die with hot spots as high as 120°C [27]. Within-die temperature fluctuations have existed as a major performance and packaging challenge for many years. Both the device and interconnect performance have temperature dependence, with higher temperature causing performance degradation. Additionally, temperature variation across communicating blocks on the same chip may cause performance mismatches, which may lead to logic or functional failures.

The net consequence of the P, V, and T variation manifests itself by chip frequency variation, resulting in a frequency distribution. This frequency distribution has serious cost implications associated with it: Low performing parts need to be discarded which in turn affects the yield and hence the cost.

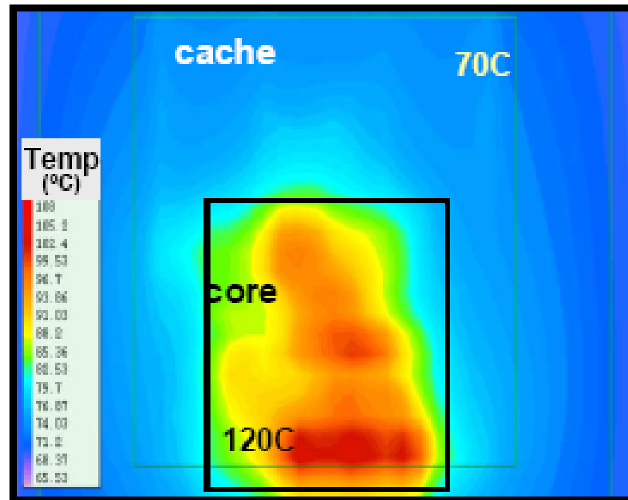


Figure 7.6: Within die temperature variation [27].

It should be noted that, as compared to logic and multicore processors, the variation of voltage and temperature (causing hotspots) in SRAM circuits is typically less pronounced, simply because SRAM circuits have highly regular structure.

Resistivity vs. Temperature

The electrical resistivity of Cu scaled-wires, which is a quantitative measure of opposition to the flow of electrical current, increases with temperature. This dependence on temperature is

usually explained with the help of a Bloch-Grneisen formula [23, 176]:

$$\begin{aligned} \rho(T) &= \rho(0) + \rho_{el-ph}(T), \\ \rho_{el-ph}(T) &= \alpha_{el-ph}\left(\frac{T}{\Theta_R}\right)^n \int_0^{\Theta_R/T} \frac{x^n}{(e^x-1)(1-e^{-x})}dx \end{aligned} \tag{7.21}$$

where the temperature independent $\rho(T)$ is the residual resistivity due to defect scattering, the temperature dependent $\rho_{el-ph}(T)$ is the due to electron-phonon interaction, n and α_{el-ph} are constants, and Θ_R is the Debye temperature (which is a measure of the hardness of the crystal).

The electrical resistivity also increases with decreasing wire widths due to surface scattering and grain boundary scattering effects [99]. As the wire width decreases from the uppermost metal layer to the lowest ll metal layer, the electrical resistivity increases. This, in turn, increases both the propagation delay time constant (τ) and the temperature rise (ΔT).

Thermal Analysis of Links

In a typical surface mount package, like IBM‘s ceramic ball grid array (CBGA), heat flows through the metal layers to the heat sink [176]. The upper metal layers have longer via (vertical interconnect) separations as compared to the lower ones. Hence the temperature rise (ΔT) in those upper metal layers is much higher and is the main cause of concern from a thermal design perspective.

Assuming that a uniform root mean square current density j_{rms} is flowing through a conductor of length L, width W, thickness H, resistivity ρ, thermal conductivity K_M, and is separated from the underlying interlayer dielectric (ILD) of thickness t_{ILD} and thermal conductivity k_{ILD}, the spatial temperature distribution along its length is given by the following

equation [176]:

$$T(x) = T_0 + \Delta T_{Max}\left(1 - \frac{cosh\left(\frac{x}{L_H}\right)}{sinh\left(\frac{L}{2L_H}\right)}\right) \tag{7.22}$$

$$\text{for} \quad -\tfrac{L}{2} \leq x \leq \tfrac{L}{2}$$

where

$\Delta T_{Max} = \frac{j_{rms}^2 \rho L^2 H}{K_M}$

$L_H = \sqrt{\frac{K_M H t_{ILD}}{K_{ILD}}\left(\frac{1}{s}\right)}$

and $\quad s = \left(\frac{w}{t_{ILD}}\left[\frac{1}{2} ln\left(\frac{w+d}{w} + \frac{t_{ILD}-\frac{d}{2}}{w+d}\right)\right]\right)^{-1}$

Each link has two vias (one at each end) and is connected to the underlying layer, which is at a temperature of T_0. The temperature of the link is actually affected by other parallel and orthogonal metal conductors separated by a spacing of d.

Since the thermal conductivity of the vias is much higher when compared to the dielectrics, heat flows rapidly through the vias to the underlying layer. The thermal model [176] of the interconnect described in Equation (7.22) incorporates the via effect and also takes the heat spreading factor (s)—which is the one dimensional heat flow from the metal wire to the underlying layer—into consideration.

Since the hottest part of a typical global interconnect is the part where the via effect diminishes, we can deduce that the probability of a hotspot lying in this area is higher. In the case where there are crossing metal bus arrays, then the probability of a hotspot lies at the intersection of those arrays.

A study by ST micro-electronics [176] performed on the spatial thermal profile of the

global Cu nano-scaled wire for on-chip interconnects in 65-nm CMOS technology, has shown that the average temperature rise ΔT due to signaling along the length of the conductor is around 6.8°C for a global interconnection link.

As the temperature increases, the interconnect delay increases due to the linear increase in electrical resistivity. This degrades the performance and shortens the interconnect's lifetime. Package reliability will also be severely affected by the resulting thermal hotspots, thus impacting the overall performance of multicore/SRAM systems. To minimize the negative impact of temperature on a multicore system running on a network of interconnects, some thermal management techniques are used.

7.2.3 PVT Variations and their Reduction Techniques

P, V, and T variations impact all levels of design. Dual-V_{th} circuit designs [27] can reduce leakage power during active operation, burn-in, and standby. Two V_{th}'s are provided by the process technology for each transistor. High-V_{th} transistors in performance critical paths are either upsized or are made low-V_{th} to provide the target chip performance. Larger transistor sizes increase the relative probability of achieving the target frequency at the expense of switching power. Increasing low-V_{th} usage also boosts the success probability, but with a penalty in leakage power. It was shown by S. Borkar et al. [27] that by carefully employing low-V_{th} devices, a 24% delay improvement is possible with a trade-off in leakage and switching power, while maintaining the same total power.

The number of critical paths that determine the target frequency varies depending on the micro-architecture design choice. Micro-architecture designs that demand increased

parallelism and/or functionality require an increase in the number of critical paths. Designs that have deeper pipelining, to support a higher frequency of operation, require an increase in the number of critical paths and a decrease in the logic depth.

Testchip measurements [27] show that as the number of critical paths on a die increases, within-die delay variations among the critical paths cause both mean (μ) and standard deviation (σ) of the die frequency distribution to decrease. This is consistent with statistical simulation results [29] indicating that the impact of within-die parameter variation on die frequency distribution is significant. So, micro-architecture designs that increase the number of critical paths will result in reduced mean frequency, since the probability that at least one of the paths is slower will increase.

Historically, the logic depth of micro-architecture critical paths has been decreasing to accommodate a 2$\times$ growth in the operating frequency every generation, faster than the 42% supported by technology scaling. As the number of logic gates that determine the frequency of operation is reduced, the impact of device parameter variation increases [27]. This is confirmed by comparing the results of the following two measurements: 1) Measurement on 49-stage ring oscillators (in 250-nm node) showed that σ of the within-die frequency distribution was 4$\times$ smaller than the σ of the device saturation current distribution [29]. However, 2) measurements on a testchip containing 16-stage critical paths (in 90-nm) showed that the σ's of within-die (WID) critical path delay distributions and NMOS/PMOS drive current distributions were comparable. In other words, the measurements show that with either smaller logic depth or with increasing number of micro-architecture critical paths, performance improvement is possible. However, the probability of achieving the target frequency that translates to performance drops

due to the impact of within-die process variation [27].

During the last decade, there has been research and design work to enhance the variation tolerance of circuits and micro-architecture. Here, we describe several significant ones including body (substrate) biasing techniques, followed by supply voltage and temperature variation tolerance methods.

A. Body Bias Control Techniques

Device performance can be improved by lowering V_{th}, however that leads to higher sub-threshold leakage current I_{sb}. One possible method to compensate for this trade-off is to apply a separate bias to critical devices.

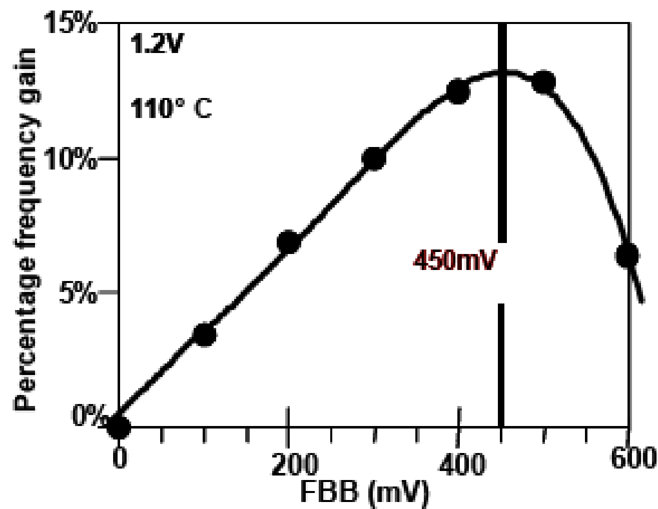


Figure 7.7: Optimal FBB for sub-90-nm generations [27].

Device V_{th} is a function of reverse body to source potential or RBB (V_{BS}). V_{th} can be modulated for higher performance by forward biasing the body (FBB). This method also reduces the impact of short channel effects, hence reducing V_{th} variations. Figure 7.7 plots the percentage frequency gain as a function of FBB. It was shown empirically that 450mV is

the optimal FBB for sub-90-nm generations at high temperature [175]. The optimal FBB for smaller nodes (e.g., 16-nm) will be less than 450mV.

In another experiment, a 6.6M transistor communications router chip [120], with forward body bias (FBB) to PMOS during active operation and zero body bias (ZBB) during standby mode, was implemented in a 150-nm CMOS technology (see [27]). FBB is withdrawn during standby mode to reduce leakage power. Compared to the original design that had no body bias (NBB), the FBB chip achieves 1GHz operation at 1.1V, compared to the 1.25V required for the NBB chip. In addition, the switching power turned out to be 23% smaller at 1GHz. Similarly, comparing the Fmax of the NBB and FBB router chips swept from 0.9V to 1.8V V_{cc} at 60°C revealed that the frequency of FBB chip is 33% higher than the NBB chip at 1.1V.

An alternate method for reducing I_{sb} is to apply reverse V_{BS}. Figure 7.8 plots the leakage current for the worst-case channel length (Lwc, dashed) and the nominal channel length (Lnom, dotted) as a function of RBB.

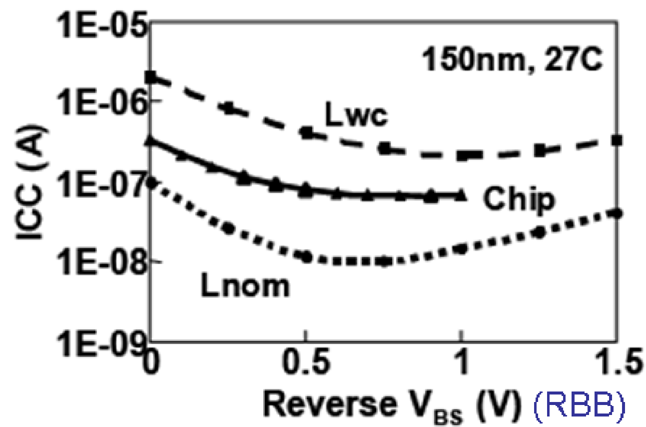


Figure 7.8: Leakage reduction by reverse body bias [27].

The measured full-chip leakage current is within these upper and lower leakage current bounds over a range of RBB values. The optimum RBB value derived from a measured 150-nm node chip for minimum leakage is 500mV [81]. Figure 7.8 suggests that using RBB values larger than a certain value (in this case, 500mV) causes the junction leakage current and overall leakage power to increase. Similarly, the effectiveness of RBB reduces as channel lengths become smaller or V_{th} is lowered. Essentially, the V_{th}-modulation capability by RBB weakens as short-channel effects become worse or body effect diminishes due to lower channel doping. This means the effectiveness of RBB will be smaller in future nodes (e.g., 16-nm) as compared to older nodes (e.g., 150-nm).

The discussion above presented the advantages of both FBB and RBB. It is possible to use both of these approaches as shown in Figure 7.9. On the left side, a typical frequency spread due to process variations is shown. The lower frequency half of the spread is too slow for optimal performance, and the higher frequency half causes power leakage. As shown on the right side, this spread can be corrected by adaptive biasing which uses both FBB (for the low frequency) and RBB (for the high frequency).

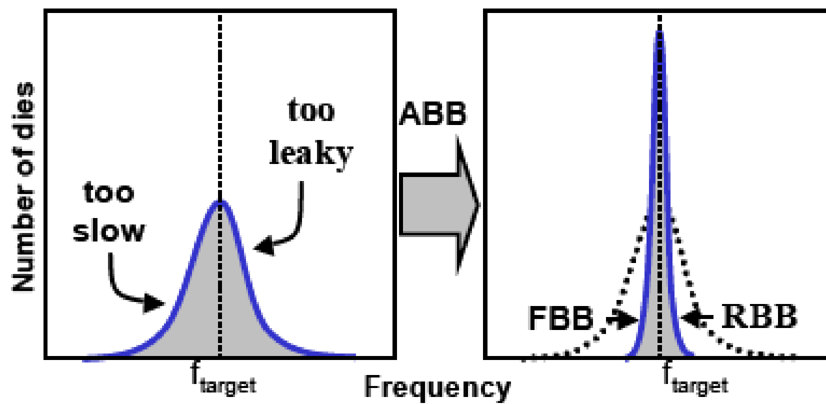


Figure 7.9: Target frequency binning by adaptive body bias [27].

According to the test results from a 150-nm CMOS technology testchip [174], by applying bidirectional ABB (used for both NMOS and PMOS devices to increase the percentage of dies that meet both frequency requirement and leakage constraint), the die-to-die frequency variation (σ/μ) is reduced by an order of magnitude and 100% of the dies become acceptable. By applying WID-ABB (multiple bias values per die to compensate for within-die as well as die-to-die variation), the σ of the die frequency distribution is additionally reduced by 50%, compared to ABB.

Such desired efficiency in applying ABB or WID-ABB, however, might not be observed in the 16-nm node, simply due to the comparatively higher σ in the smaller nodes.

B. Supply Voltage and Temperature Control Techniques

As mentioned earlier in this section, variations in switching activity and diversity of the type of logic result in uneven supply voltage distribution and temperature hot spots across the die. A technique to increase yield in the high frequency bins is to apply adaptive V_{cc}, as was confirmed by experimental results on a 90-nm test chip [27]. It was observed that by applying adaptive V_{cc} (instead of fixed V_{cc}) more than 20% of dies could move from the lowest acceptable frequency bin to the more desirable higher frequency bin [27].

Adaptive V_{cc} does not solve the ΔV_{cc} problem. A well-known technique to mitigate voltage variation, namely, adding on-die decoupling capacitors, is explained in Section 7.3.

Maximum temperature and within-die temperature variations have to be controlled. Throttling is used to control the die temperature, and Figure 7.10 explains the method. When the temperature rises above the maximum limit set by the frequency and power, the operating frequency is lowered, followed by the die V_{cc}. Subsequently, the power consumption drops

which lowers the on-die temperature. When the die comes out of power saving mode, V_{cc} is raised followed by frequency. Many commercial processors incorporate throttling.

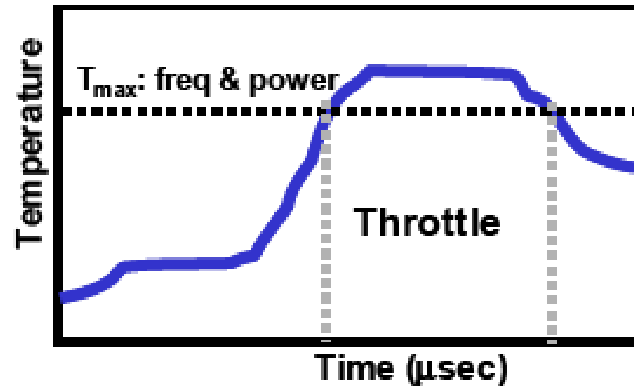


Figure 7.10: Temperature based V_{cc}/frequency throttling [27].

A number of methods have been proposed to create temperature-insensitive designs, either taking advantage of the temperature-insensitive voltage [191] or by adjusting the threshold voltage to achieve temperature insensitivity [93]. Additional approaches to achieve temperature-insensitivity include the use of multiple threshold designs to balance the dependences of high-V_{th} and low-V_{th} logic cells [32]. Another approach suggests taking advantage of the low swing voltages and temperature-aware system design to improve system energy while limiting the impact of temperature variation. Furthermore, there have been some studies aimed at the minimization of process, voltage, and temperature variations (PVT) all together.

One such study [172] has proposed tunable replica circuits (TRC) to improve the traditional processor design that build margins into operation voltage (V) and frequency (F) to account for V, T, and aging variations in an attempt to ensure error-free operation under worst-case conditions. The traditional processor design fails to exploit the opportunities for V-F

improvement during favorable V-T conditions and lack of aging degradations.

By proposing TRC (used in conjunction with error-detection sequentials (EDS) and dynamic voltage and frequency techniques), this study illustrates how to mitigate V, T, and aging and how to exploit V-F improvement opportunities dynamically. The measurements on a 45-nm test chip from this study demonstrate the delay sensitivity of replica circuits to voltage, temperature, and AC/DC stress and recovery.

In one of the results from this study—released by an Intel research group—Keith Bowman et al. illustrate the delay sensitivity of the key components of today‘s scaled-down circuits (e.g., SRAM, BUS, INV, NAND, NOR, etc), to supply voltage and temperature changes. Figure 7.11(a) shows the delay change due to the voltage droop (the intentional loss in output voltage from a device, as it drives a load, to increase the headroom for load transients) for two different supply voltages of 1.15V and 0.85V for several different components implemented in 45-nm technology. It is shown that a voltage droop of 150mV from the supply voltage of 1.15V leads to a 15% delay increase in a BUS and a 20% delay increase in an inverter gate. For the supply voltage of 0.85V, however, the same droop of 150mV leads to a 40% delay increase in the same BUS and a 65% delay increase in the same inverter gate. The considerable (more than a factor of 2) difference between the delay increases clearly implies that the impact of V_{dd} variation is more dramatic for circuits with lower supply voltages.

Figure 7.11(b) shows the delay change due to a temperature change for two different supply voltages of 1.15V and 0.7V for several different components implemented in 45-nm technology. From this figure, we observe that whereas a temperature increase of 40°C (from 50°C to 90°C) for the supply voltage of 1.15V leads to a 4.5% delay increase in a BUS and

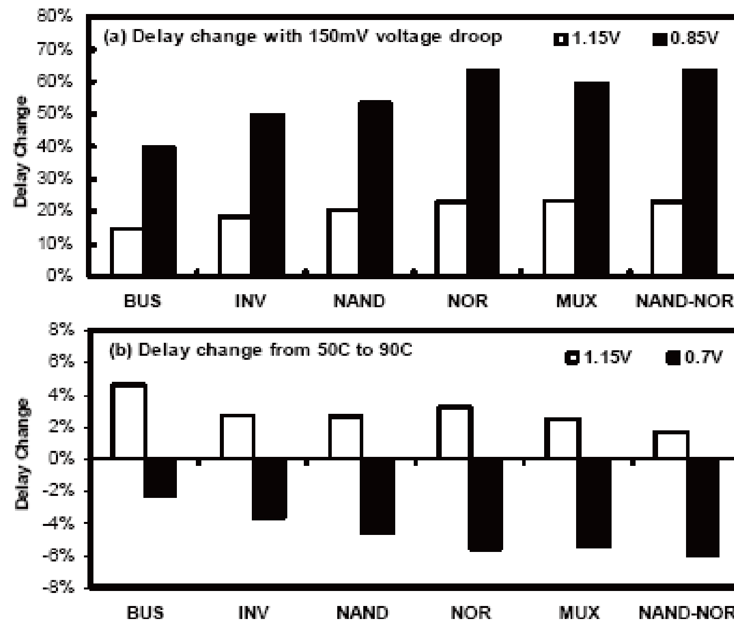


Figure 7.11: Measured delay changes to V_{cc} and Temperature [172].

a 3% delay increase in an inverter gate, the same temperature increase of 40°C but with a supply voltage of 0.7V leads to a 2.2% delay reduction in the same BUS and a 3.5% delay reduction in the same inverter gate. Figure 7.11(b) illustrates an interesting point. While for the higher supply voltage of 1.15V the delay is increased by an increase in temperature, for the low-voltage of 0.7V, the delay is decreased by an increase in temperature. This phenomenon is called "reverse temperature dependence" [191], which shows up only in low-voltage designs: [191, 172, 73]. It is the opposite of the "normal temperature dependence" phenomenon, in which the delay is increased when temperature is increased.

The notion of "reverse temperature dependence" has inspired some network on chip (NoC) designers to consider applying reduced swing voltages to the interconnect links in an effort to reduce power consumption of the chip [191].

Unfortunately, the reduction of the link voltage makes these systems much more susceptible to variation. Temperature variations have a particularly severe impact on delay in low voltage designs, shown in Figure 7.12 for a 65-nm commercial technology. The error bars in Figure 7.12 are quantified in Table 7.2, where we see that the military specified temperature range (-55°C to +125°C [124]) can result in delay changes in excess of 200% at 0.6 V. Another important observation from Table 7.2 is that the delay change from -55°C to 125°C is negative at 0.6 V and positive at 1.0 V

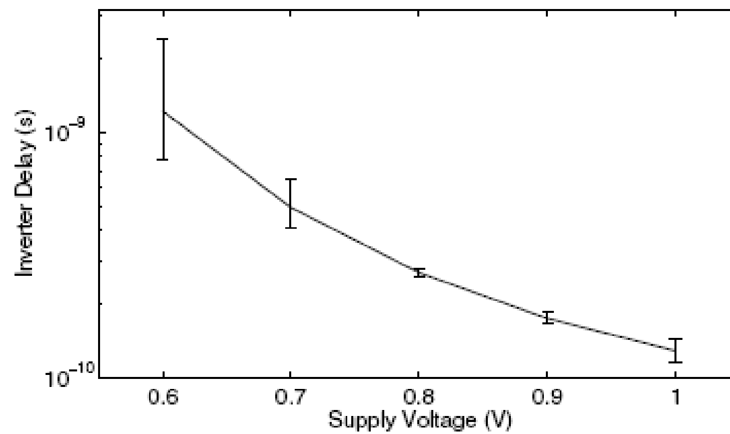


Figure 7.12: Impact of temperature on a commercial 65-nm technology [191].

Table 7.2: Temperature-induced delay change in a 65-nm technology [191].

Compared Temperatures	Link Voltage				
	0.6 V	0.7 V	0.8 V	0.9 V	1.0 V
-55°C→125°C	-210.0%	-59.3%	-8.2%	11.4%	19.3%
25°C →125°C	-59.6%	-22.2%	-4.5%	4.5%	8.9%
45°C→125°C	-42.2%	-16.5%	-3.5%	3.3%	6.8%
65°C →125°C	-28.4%	-11.5%	-2.7%	2.23%	4.9%

The voltage region with a positive delay change (normal temperature dependence re-

gion) is on the right of 0.8V, while the region with a negative delay change (reverse temperature dependence region) is on the left side of 0.8V. Between the two regions, there is a supply voltage where the impact of temperature dependence on delay is minimized [191], as indicated by the smallest error bar at 0.8 V in Figure 7.12. This is referred to as the temperature-insensitive voltage $VINS$, and as technology scales down, this voltage approaches the nominal voltage [190].

The difference between the temperature dependence at high and low voltages provides an interesting opportunity for systems with reduced link swing voltages: if the link voltage is low enough to operate in the reverse temperature dependence region, a change in temperature will cause the link delay to vary in the opposite direction of the delay in the nominal voltage router. These opposing delay variations make room for innovative approaches to lessen the impact of temperature variations and improve system reliability and performance.

In a recent work, D. Wolpert et al. [191] has proposed a temperature-aware delay borrowing method that averages the impact of temperature variation on the link and the transceiver. This proposed work shows that when the links are operated below $VINS$, the average of the reverse temperature dependence in the link and the normal temperature dependence in the transceiver results in a large reduction in the impact of temperature variation on the communication system as a whole.

The above discussion, however, does not necessarily mean that the impact of temperature on delay will be reduced in all future circuits due to VINS approaching the nominal voltage. Only some selected circuits that can borrow delay in their critical path—such as a system composed of a transmitter router, a receiver router, and a link with a temperature sensor in between, illustrated by D. Wolpert et al. [191]—may benefit from the "reverse temperature

dependence" phenomenon.

Due to having a very regular and dense structure, and therefore not having too many interconnects (especially, as compared to multi-core processors), typical SRAMs cannot use delay borrowing techniques for energy-efficiency purposes without compromising their speed.

However, there have been other proposed techniques to help minimize the adverse impact of temperature and voltage variation on SRAMs, with insignificant loss of performance. One such study [143], co-authored by K. Boman and his reputable research group, proposes the use of tunable replica bits (TRBs) for mitigating a part of the V_{cc} guardband (V_{cc} GB). This study was inspired by infrequent dynamic events like V_{cc} drops and temperature changes that naturally result in the use of a static V_{cc} GB in 8T-SRAM arrays.

The 8T-SRAM cell (Figure 7.13) is commonly used in single-V_{cc} micro-processor cores for performance-critical low-level caches and multi-ported register-file arrays [94]. The 8T-cell offers fast read (RD) and write (WR), dual-port capability, and generally supports lower minimum V_{cc} (or V_{MIN}) than the 6T-cell at the expense of extra area, among others. By using a decoupled single-ended read (RD) port with a domino-style hierarchical RD bit-line, the 8T-cell features a fast RD evaluation path without causing access disturbance that limits RD V_{MIN} in the 6T-cell. Using the 8T-cell in a half-select-free architecture eliminates pseudo-reads during partial writes, hence enabling write (WR) V_{MIN} optimization independent of RD.

As power limits are aggressively reduced, the demand for a lower V_{MIN} in the 8T-cell is increased. Similarly, as feature size is aggressively reduced, the variation of both within-die (WID) and die-to-die (D2D) device parameters worsen. The typical approach of sizing up the 8T RD and WR ports to mitigate process variation has limited V_{MIN} returns. In the RD case,

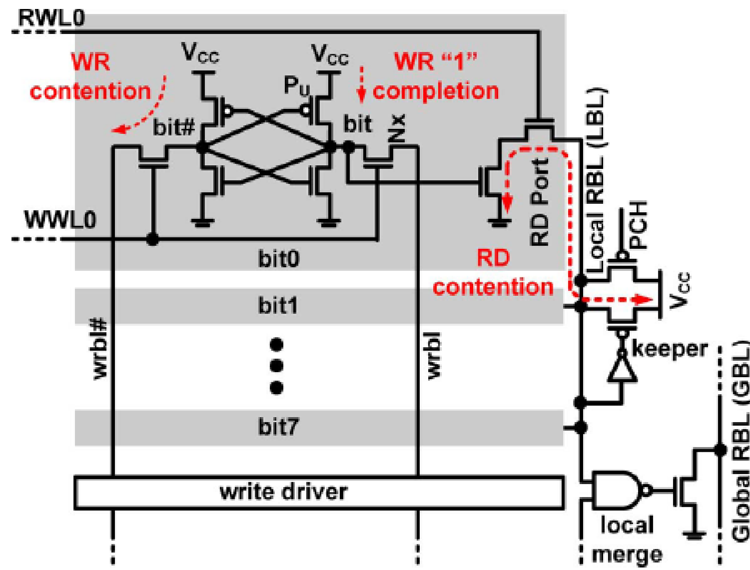


Figure 7.13: The 8T-SRAM cell architecture showing the WR and RD ports [143].

using a larger nMOS RD port helps when reading a "1" by reducing contention with the pMOS domino keeper on the local bit-line (LBL). To compensate for the degraded noise margin that comes with a larger (and leakier) RD port, the pMOS keeper needs to be relatively upsized for good reading of a "0," resulting in diminishing V_{MIN} returns with continued upsizing. Similarly, contention between the pMOS pull-up (PU) and the nMOS pass (N_X) impacts WR V_{MIN} and is reduced by sizing up N_X. However, P_U needs to be relatively sized up as well, since at a given point a weak P_U will limit the completion of write within a given WL pulse. Consequently, various RD as well as WR assist techniques for lowering the static V_{MIN} of 8T-arrays are being investigated [84].

In spite of careful design and optimization of such techniques, a constant V_{cc} GB is always employed to ensure functionality under highly infrequent dynamic events. Hence, the elimination of the V_{cc} GB (or at least a part of it) is crucial to reduce the final V_{MIN} of an

8T-SRAM array and any core logic that runs on the same supply.

The TRB provides a monitor (special circuit) which indicates if an access error has occurred in the memory under a dynamic event. This monitor, in conjunction with a system level recovery technique, can ensure correct functionality even under such events. Different techniques to recover once an error has been detected have been reported by J. Tschanz et al. [173]. These techniques include, but are not limited to, replaying the instruction at the same CLK frequency or replaying at slower (e.g., half) CLK frequency. Further, an error counter can be employed to track the number of errors over a period of time. A large error count (at the expense of an increased recovery penalty) indicates that under the given environmental and/or process conditions, a higher V_{MIN} is required. Consequently, a dynamic voltage/frequency scheme (DVFS) can be employed to temporarily increase the operating V_{MIN}. The details of these techniques and their relative merits and demerits are beyond the scope of this thesis; interested readers are directed to [173].

In brief, in the TRB scheme, TRBRD (TRBWR) generates error signals whenever the supply voltage drops below VTRBRD (VTRBWR), plus a small TRB margin:

VTRBRD = V_{MIN}RD + TRB Margin for RD,

VTRBWR = V_{MIN}WR + TRB Margin for WR.

where TRBRD and TRBWR are read TRB and write TRB circuits and VTRBRD and VTRBWR are the static RD and WR V_{MIN}.

To evaluate the impact of such first-order voltage droops on the array, a plot of the measured number of single bit failures (SBF) as a function of V_{cc} without any induced droop

and with A_{DROOP} = 13% is plotted in Figure 7.14. To capture the effect of the maximum droop on RD and WR operations, the capture of the error signal is synchronized with the worst case droop condition. From Figure 7.14 it can be observed that:

(a) RD is the V_{MIN} limiter for the given array at a target frequency of 1 GHz (VTRBRD $\geq$ VTRBWR, both with and without droop), and

(b) a V_{cc} GB of 9.5% for RD (and 10.2% for WR) is necessary for error-free array operation in the presence of voltage droops (ADROOP = 13%).

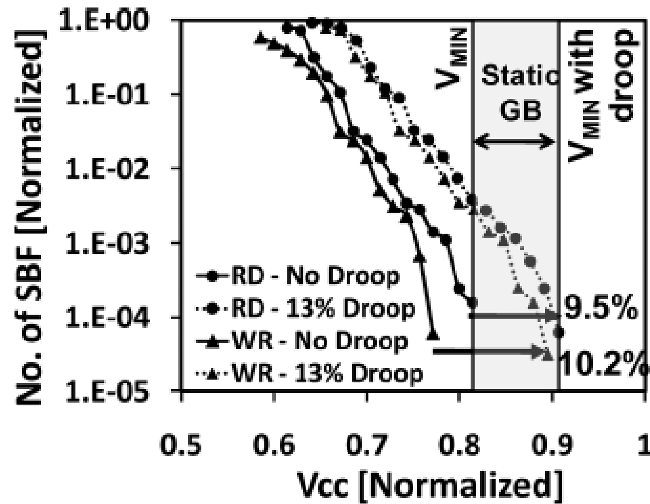


Figure 7.14: Measured number of single bit failures in the 16 KB array with and without V_{cc} droop (A_{DROOP} = 13%). A 9.5% static guardband (GB) is required [143].

Both the TRBRD and TRBWR need to be calibrated in such a way that whenever there is an actual failure in the array, the TRBRD must fail. The additional small TRB margin ensures that the TRB fails just before any actual failure. In the experimental setup [143], the VTRBRD and VTRBWR are set with a TRB margin of 20 mV above V_{MIN}RD and V_{MIN}WR and the operating V_{cc} is pushed to VTRBRD for a maximum power benefit. Whether or not

an actual array failure occurs, whenever an Error signal is generated by the TRB, the set up assumes that there has been a failure in the array and corrective measures (e.g., increasing V_{MIN}) are employed. The use of the TRB enables the designer to mitigate a part of the static V_{cc} GB and ensures an operating array V_{cc} = VTRBRD.

Because the TRB is designed to generate an error signal whenever the array encounters a V_{cc} droop, the performance overhead of this technique depends on the number of error corrections which must be performed, and hence scales with voltage droop rate.

The experimental results [143] obtained from the measured error rates for RD and WR, assuming a worst case 13% V_{cc} droop every 100 ms, show that even with such a high droop rate, the error rate is 10% when operating at VTRBRD—which is expected to cause less than 1% net performance degradation [166].

The use of TRBs incurs additional overhead in terms of area (expected to be very small for a typical array size) and power. This power consumption occurs mainly in the tuning circuits of the TRBs. At high V_{cc}, the TRBs show a 20% increase in operating power. However, the use of TRBs coupled with error recovery allows reduction in operating V_{MIN} by $\sim$9%. Also, the measured power of the array (at an access rate of 10%) reveals that as the V_{cc} scales, the power overhead of the TRB is amortized and results in a 7.5% net power gain for the array alone.

We will see the effect of temperature and voltage variation on 16-nm 6T-SRAM in Chapter 10 (simulation).

7.3 IR-Drop, EM, and Ldi/dt

Advances in process technology and changing design styles are increasing the impact of IR-drop, electromigration (EM), and Ldi/dt effects on the performance and reliability of analog, mixed-signal, memory and custom digital IP blocks at 28-nm and below [54].

The power distribution network connects the power and ground voltages from the pad locations to the devices in a circuit. Shrinking device dimensions, faster switching frequencies, and increasing power consumption in deep sub-micrometer technologies cause large switching currents to flow in the power and ground networks, degrading performance and reliability. A robust power distribution network is essential to ensure reliable operation of circuits on a chip [46].

The resistivity of most conducting metals increases linearly with temperature due to the Joule heating by electrical currents flowing through the conductors. Therefore, in order to accurately characterize the performance of high-power integrated circuits (ICs), packages and printed circuit boards (PCBs), it is essential to account for both electrical and thermal effects and the intimate coupling between them [155].

Power supply integrity verification is a critical concern in high-performance designs, as well. Due to the resistance of the interconnects, there is a voltage drop across the network [46]. The amount of DC change in the power supply voltage is usually referred to as the "IR-drop." The IR-drop is a function of the average value of the current that the circuit draws from the power supply network, which randomly varies temporally and spatially. As a result, the spatial variation of the IR-drop across power distribution network is usually considered un-

predictable. Additionally, power saving techniques, e.g. clock gating and sleep transistor logic, tend to increase the variability of spatial and temporal IR-drop distributions across the chip [85].

The package supplies current to the pads of the power grid either by means of package leads in wire-bond chips or through *C4 bump arrays* in flip chip technology. Although the resistance of the package is quite small, the inductance of the package leads is significant, which causes a voltage drop at the pad locations due to the time-varying current drawn by the devices on the die. This voltage drop is referred to as the *di/dt-drop*. Therefore, the voltage seen at the device-level is the supply voltage minus the IR-drop and *di/dt-drop* [46].

Excessive voltage drops in the power grid reduce switching speeds and noise margins and inject noise which might lead to functional failures. High average current densities lead to undesirable wear of the metal wires due to electromigration (EM). Therefore, the challenge in the design of a power distribution network is in achieving excellent voltage regulation at the consumption points while considering the wide fluctuations in power demand across the chip, as well as minimizing the area of the metal layers. These issues are prominent in high performance chips such as microprocessors (using SRAMs), since large amounts of power have to be distributed through a hierarchy of many metal layers. A robust power distribution network is vital in meeting performance guarantees and ensuring reliable operation. Looking from a general technology perspective, there are three distinct trends contributing to the significance of EM and IR-drop effects in modern IC designs: 1) Driven by Moore‘s law, metal interconnect widths are decreasing exponentially. As a result, the overall cross-sectional area of the interconnect is shrinking. 2) With increasing integration of functionality and passive devices, the total interconnect length is exploding as well, which means that there are more wires on a die

that are susceptible to EM effects. 3) Currents are not scaling proportionally to shrinking wire widths, and therefore, modern ICs have extremely high current densities.

There are **two different EM failure mechanisms** that occur due to asymmetry in the ion flow: 1) open circuit and 2) hillock, as illustrated in Figure 7.22 (see Section 7.4, Interconnect). Open circuit failure occurs in the interconnect where the outgoing ion flux exceeds the incoming ion flux. Hillock failure occurs in the interconnect when the incoming ion flux exceeds the outgoing ion flux [54]. In addition to these two physical failure mechanisms, there is a temperature-induced failure mechanism known as *cyclical positive feedback loop* that ultimately ends in failure [54].

Once a void begins to develop in a metal wire, the wire itself becomes narrower at that point. Due to the reduction in width, the current density increases and the interconnect temperature increases due to Joule heating. Joule heating is a result of root-mean square (RMS) current. As the temperature of the wire increases, the growth of the void accelerates, and eventually an open circuit occurs. This is why it is critical to also take RMS current into account when performing EM analysis.

One solution to mitigate the EM effect is to increase the wire width to reduce the current density. Another solution, complementing the first solution, is to use some layout strategies for interconnects such as non-90° corners, non-rapid wire width reduction, and via arrangement strategies to avoid current crowding and a rapid increase in current density.

The capacitance between the power and ground distribution networks, referred to as the decoupling capacitors or *decaps*, acts as local charge storage and is helpful in mitigating the voltage drop at supply points. Parasitic capacitance between metal wires of supply lines,

device capacitance of the non-switching devices, and capacitance between N-well and substrate occur as **implicit decoupling capacitance** in a power distribution network. Unfortunately, this implicit decoupling capacitance is sometimes not enough to constrain the voltage drop within safe bounds and designers often have to add intentional **explicit decoupling capacitance** structures at strategic locations. These explicitly added decoupling capacitances are designed for a high capacitance-to-area ratio; therefore, they are not free and increase the silicon area and leakage power consumption. Parasitic interconnect resistance, decoupling capacitance, and package/interconnect inductance form a complex RLC circuit which has its own resonance frequency. If the resonance frequency lies close to the operating frequency of the design, large voltage drops can develop in the grid.

Decap layouts are designed for a high capacitance-to-area ratio, however, that tends to increase the gate oxide area. The oxide leakage and area penalty must be compensated with ΔV_{cc}. There exists several research works, as well as several IR-Drop and EM Ldi/dt analysis tools, that help with such trade-off analyses and accurately characterize the performance of ICs and PCBs.

For example, while some advanced packaging research discusses various technologies, such as design, thermal management, design for test, fabrication, and system integration technologies, three-dimensional stacking technology provides a flexible strategy to achieve extremely high interconnect densities (by chip stacking) and offers an opportunity to go “beyond Moore‘s law” [155].

The current flowing in the resistive copper foils—that make up a power delivery network (PDN) for a chip—produces two detrimental impacts, namely IR-drop and an increased

temperature. These two issues are getting worse as the newer technology nodes tend to use lower operating voltages. The reason for this exacerbation of IR-drop and rising temperature is that the power-hungry devices, whilst operating at low voltage, can draw considerable current and generate more heat at the same time. More IR-drop, in turn, means a lower voltage available at the load, which could lead to functional failure. A higher temperature, in turn, means a higher possibility of damaging the laminate of the PCB and ultimately melting the copper foil.

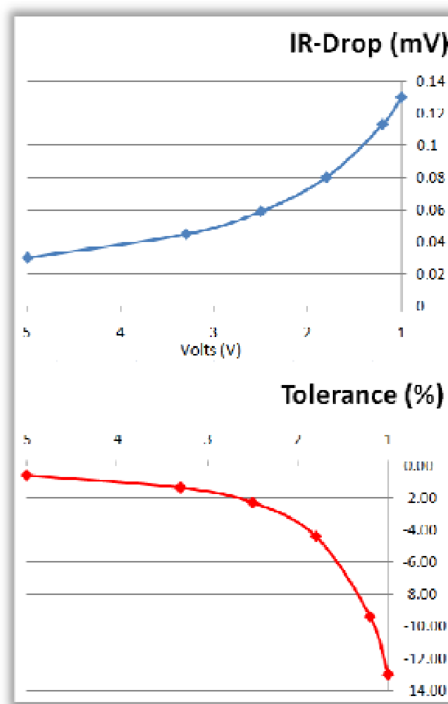


Figure 7.15: IR-Drop & Tolerance vs. V_{dd} [62].

In one recent report [62], "IR-Drop Analysis," C. Halford (Advanced Layout Solutions Ltd) quantifies the increased amount of IR-drop in systems using lower voltages. He considers a 10W nominal load device that is supplied via a 15mΩ PDN to quantify the IR-drop. According to his experimental results (Figure 7.15), while in a 3.3V system the DC voltage drop

would only be 45mV (1.4%) (which can generally be considered an acceptable loss), in a 1.2V system, the DC voltage drop would be nearly 10%. In a 1.0V system, the loss is even worse (14%). Figure 7.15 shows that the loss increases dramatically for lower voltage devices of the same power dissipation and the lower voltage systems are far more sensitive to the quality of the power connections designed into the board.

Figure 7.16 shows a well-known technique to mitigate voltage variations (IR-Drop, Ldi/dt), namely adding on-die decoupling capacitors in sub-90-nm technology. ΔV_{cc} reduces from 15% to less than 10% by carefully placing the appropriate amount of decaps on microprocessor dies [142].

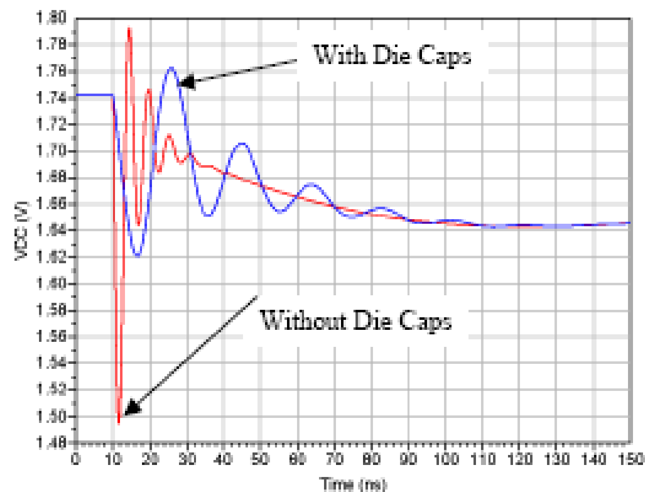


Figure 7.16: Effectiveness of on-die decoupling capacitors [27].

Another way to ensure that IR-drop and temperature problems are avoided is to use wide planes of thick copper for all power connections. Unfortunately, routing space is usually at a premium and unnecessarily large power connections will inevitably compromise the design in other ways. Using thick copper can have its drawbacks too, as it is incompatible with fine trace

widths used on ‘mixed‘ signal/power layers. Although the calculations required to estimate IR-drop are not complex, actually carrying out the task manually is daunting. The reason for this is that the copper shapes usually end up being very complicated. In both plated-through-hole and microvia technology, holes are formed around the vias as they pass through the power planes. This “Swiss cheese” effect can seriously compromise the performance of a power plane, and makes manual calculation very awkward.

Over a distance of copper, there is also a static voltage drop that must be accounted for in the system power supply budget. In addition, the AC voltage fluctuations must also be considered.

A recent thermal-aware IR-drop analysis study [155] uses the temperature profile from a steady-state thermal analysis to update the power grid resistance for IR-drops at each simulation loop, as shown in Figure 7.17. To elucidate the issues involved, this study considers an elaborate chip package-PCB model introduced by Xie et al. [69] and enhanced by Y. Shao et al. [155]. The latter includes heat sinks, thermal interface materials, through silicon vias (TSV), a controlled collapse chip connection to spread the heat from high heat flux areas, multiple chip modules, and a PCB.

Presented in Figure 7.17 is a flow diagram for the thermal-aware IR-drop co-analysis. First, the IR-drop/conduction analysis is performed at room temperature for the initial voltage distribution. Subsequently, the computed power dissipation from DC voltage leakage is imposed as the heat source for steady-state thermal analysis. Moreover, the power profiles $P = J \cdot E$ of the chip and memory modules are included as heat sources in the thermal temperature computation. Once the temperature distribution within the device is calculated, it can be used to update

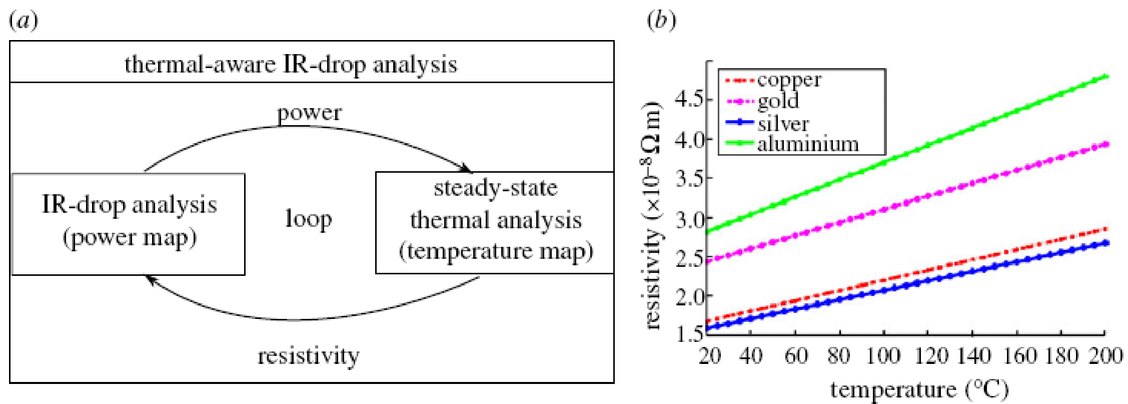


Figure 7.17: Electrical-thermal coupling. (a) Flow chart and (b) temperature-dependent resistivity of metals [155].

the material properties, specifically the conductivity (resistivity) of the conducting materials. As previously discussed, electrical resistance has a linear temperature dependence (Figure 7.17(b)). For instance, when the temperature of the device increases from room temperature to 80°C, the electrical resistivity will increase by more than 40%. Consequently, the updated values of the resistivity within the device lead to a modified voltage distribution. The fully coupled multi-physics, electrical, and thermal analysis is repeated until the temperature-dependent IR drop and thermal distribution converge with negligible error.

Figure 7.18 illustrates that the IR-drop and the temperature rise (which is due to resistive power loss I^2R) can be displayed in existing high-tech tools such as the IR-drop analysis tool built into Allegro PCB SI^{TM} v16.x. The gradual change in color (Figure 7.18) reveals the local pockets of high current-density, in which a high risk of excessive heating could exist [62]. The IR-drop analysis tool can highlight potential problems in power delivery paths, providing visibility for both IR-drop and “hot-spotting” issues. The tools provide a basis for correctly designing high-current power connections by quantifying the amount of voltage drop and tem-

perature rise that are to be expected, thus deterring the over-engineering that often accompanies uncertainty. The temperature analysis is particularly useful for ensuring that a sufficient number of parallel vias have been used in the power paths.

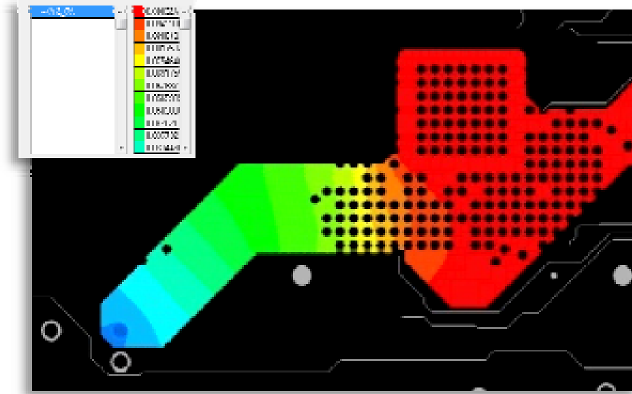


Figure 7.18: Voltage Drop on Plane Shape [62].

In Chapter 9, we incorporate the effect of IR-drop and Ldi/dt variation in our 6T-SRAM design by assigning a rather small sigma for D2D and WID variations of V_{dd} in our first-order statistical modeling.

7.4 Interconnect Challenges

The scaling of digital electronic device dimension and performance has been driven by Moores' law over the last 40 years. As a result, logic components in a microprocessor have shown dramatic performance improvement. On the other hand, an on-chip interconnect, which was considered only as a parasitic load before the 1990s, has become the real performance bottleneck due to its extremely reduced cross-section. Now, an on-chip global interconnect, made with a conventional Cu/low-k material and a delay-optimized repeater scheme, faces great

challenges in the nanometer regime, imposing problems of slower delay, higher power dissipation, and limited bandwidth. **Carbon based materials**—such as *carbon nano-tubes* (CNT) and *graphene nano-ribbons* (GNR)—and **optical interconnects** have been proposed as solutions for future nodes due to their special physical characteristics [65, 87].

Therefore, due to its crucial relevancy to the future technology nodes, this section describes the basic physical properties of Cu/low-k interconnects (used in our SRAM design), and compares their attributes to those potential alternatives (such as carbon-based and optical interconnects). Following the introductory discussion of these different interconnects, this section will then illustrate the performance of these interconnects, as well as their virtues and drawbacks. We start with a brief overview of the role of interconnects in chips followed by a brief list of the requirements of the interconnect materials in the next two sections, respectively.

7.4.1 Overview of Interconnect

Once the active devices and regions are fabricated, they must be electrically connected to make circuits. They must also be connected to the outside world through their inputs and outputs on bonding pads. Making these connections is the job of contacts, vias and interconnects. Dielectric layers are used to separate the interconnects from one another. All of these components are part of the "metallization" or "backend" structure. Figure 7.19 is a simplified schematic diagram showing these components in a typical integrated circuit structure. In recent times, the relative importance of the backend structure has greatly increased, and will likely continue that way as integrated circuit technology progresses.

Interconnects can either be *global*, *semi-global*, or *local*. In general, *local* intercon-

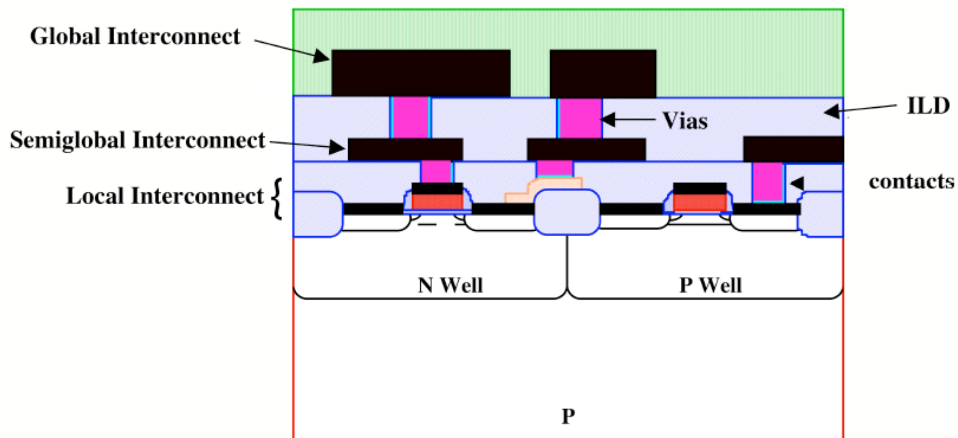


Figure 7.19: Schematic cross-section of backend structure, showing interconnects, contacts, and vias, separated by dielectric layers [148]. ILD stands for interleaved dielectric.

nects are the first, or lowest, level of interconnects. They usually connect gates, sources and drains in MOS technology, and emitters, bases, and collectors in bipolar technology. *Global* interconnects are generally all of the interconnect levels above the *local* interconnect level. They often travel over large distances, between different devices and different parts of the circuit, and therefore are typically low resistant metals. *Semi-global* interconnects are placed in between the *global* and *local* interconnect, and have the weaker attributes of the *global* and *local* interconnect. The hierarchy of interconnects can be summarized as shown below:

- *Local* interconnects—used for very short interconnects at the device level.
- *Semi-global* interconnects—used to connect devices within a block.
- *Global* interconnects—used to connect long interconnects between the blocks, including power, ground and clocks.

7.4.2 Requirements of the interconnection materials

Below is a summary of the requirements of the interconnect materials. The characteristics and performance of the most important interconnect materials will be discussed in the following sections.

- Low resistivity of conductors
- Low capacitance ⇒ low dielectric constant
  - Low RC delay
  - Low cross talk
  - Low power dissipation (CV^2f loss)
- Low inductance
- Resistance to electromigration
- Ease of deposition of thin films of the material
- Ability to withstand the chemicals and high temperatures required in the fabrication process
- Ability to be thermally oxidized
- Good adhesion to other layers—low physical stress
- Stability of electrical contacts to other layers
- Ability to contact shallow junctions and provide low resistance

- Good MOS properties
- Ability to be defined into fine patterns—dry etching
- Good large mean free path (MFP), I_0—for carbon nano-tubes (CNT)
- Good low packing density (*PD*)—for CNT
- Low modulator capacitance (C_{mod})—for optics
- Low detector capacitance (C_{det})—for optics
- Low wire segment resistance (R_w) and capacitance (C_w)—for capacitively driven low-swing interconnect (CDLSI)
- Good large bisectional bandwidth density—for CDLSI
- Good systems metrics such as high bandwidth density with low power density and low latency
- Low power dissipation during high switching activity (SA)

7.4.3 Progress Trend and Future of Interconnect

During the last few years, Network-on-Chip (NoC) architectures have gained popularity to address the interconnect delay problem for chip multi-core (CMP) systems on chip (SoC) in deep sub-micron technology. However, both two dimensional (2D) and three dimensional (3D) NoC designs still face challenges involving high performance and energy-efficient interconnects.

Recently, by stacking active silicon layers, 3D ICs have achieved a number of advantages over that of the traditional 2D design [130]: (1) shorter *global* interconnects; (2) higher performance; (3) lower interconnect power consumption due to wire-length reduction; (4) higher packing density and smaller footprint; and (5) support for the implementation of mixed-technology chips. In this context, several 3D designs, from distributing different logical units among different layers to splitting a unit (such as a processor) into multiple layers have appeared [139]. However, 3D stacking may result in temperature hotspots due to increased power density. The increased temperature in 3D chips has a negative impact on performance, leakage power, reliability, and the cost of cooling. Thus, any 3D design should consider the thermal issues in addition to other design parameters.

Figure 7.20 shows a 3D stacked NoC router architecture, which is stacked into multiple layers and optimized to reduce the overall area requirements and power consumption.

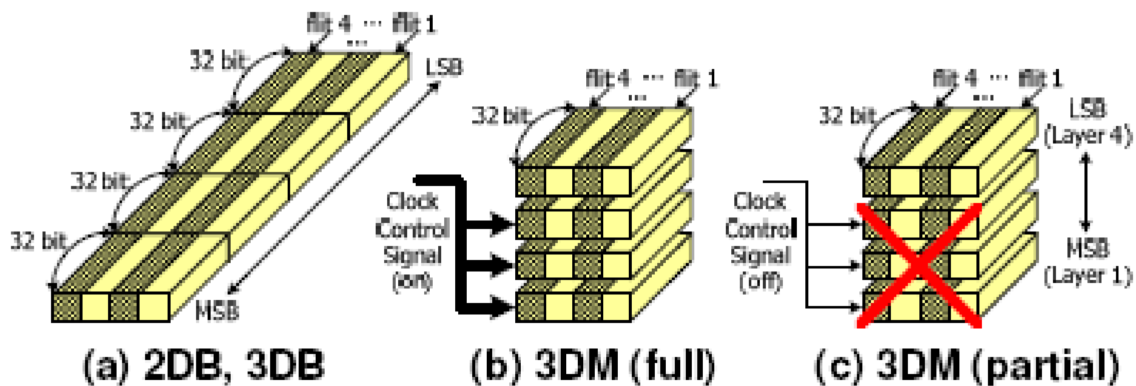


Figure 7.20: Input Buffer Distribution. (a) Basic single layer (b) Multi-layer with all layers active (c) Multi-layer with the bottom three layers shut down to save power and prevent hot spots [130].

Figure 7.20(a) shows that the width (W) of each flit (or block) has been divided into four 32-bit segments, with the least significant bit (LSB) in the first and the most significant bit

(MSB) in the last segment. Figure 7.20(b) and Figure 7.20(c) show that the wordlines of the buffer span across *L* layers, while the bitlines remain within a layer. For example, if W=128 bits and L=4 layers, then each layer has 32bits starting with the LSB on the top layer and MSB at the bottom layer. Such NoC architecture allows one to selectively power down the bottom layers of a multi-layer NoC that have redundant or no data (all 0 word or all 1 word or short address flits: a block that has redundant data in all the other layers except the top layer of the router data-path), helping in energy conservation and partially mitigating the thermal challenges in 3D designs.

Although chips/SRAMs can benefit from clever NoC architecture designs to boost their efficiency, they can achieve even higher performance goals by using more advanced interconnects.

The ever-decreasing cross-section of the interconnect has given rise to an increase in resistance. In addition, the surface and grain-boundary scattering of electrons in copper (Cu) has become a serious problem as the wire size has become almost comparable to the grain size of Cu, eventually leading to higher resistivity than the bulk [163, 75]. Putting all these together, degradation of the RC time constant of on-chip metal wires has become more serious. As a result, the continuous performance degradation of on-chip Cu/low-k interconnects is one of the greatest challenges in keeping Moore‘s law alive while the scaling of the transistor‘s dimensions have provided relentless delay improvement as shown in Figure 7.21.

The scaling of the wire dimensions deteriorates not only the delay time but also all related interconnect performance metrics, such as power dissipation, reliability, and bandwidth for *local*, *semi-global* and *global* interconnects. The on-chip power dissipation problem is made

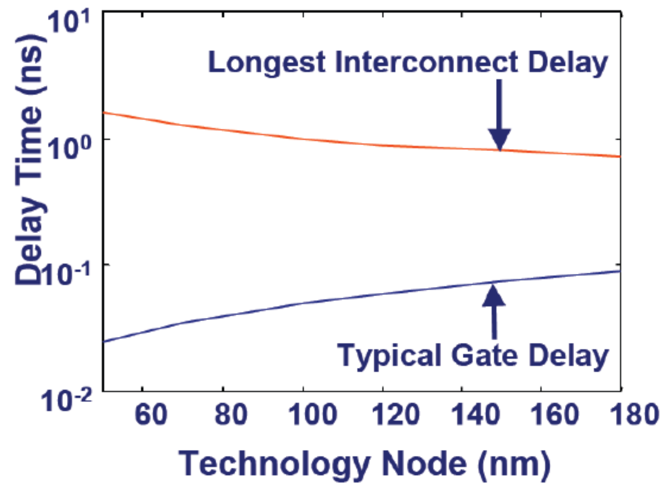


Figure 7.21: Delay as a function of technology node both for *global* interconnect and typical CMOS gate [87].

worse by the increasing number of repeaters used to alleviate the long RC time constant of a Cu wire, switching activity factor, and increase of operating frequency. The reliability issue also becomes very important since future systems will require higher current density within the reduced wire cross-section to maintain or boost the operating frequency. This is directly related to electromigration-induced hillocks and voids in Cu as shown in Figure 7.22.

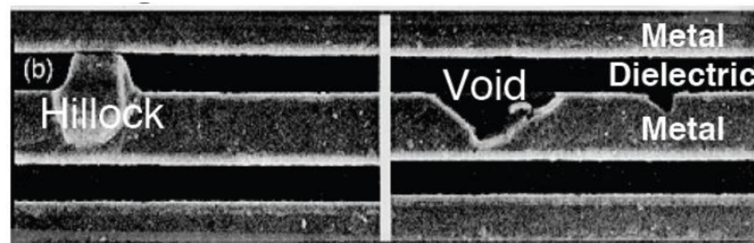


Figure 7.22: Hillocks and voids induced by electromigration with high current density in a Cu interconnect [87].

Both hillocks and voids are detrimental to on-chip signaling because they are responsible for shorts between adjacent interconnect lines and opens to single signal paths which are

the main causes of functional failure [37].

Therefore, it has become imperative to investigate novel interconnect technologies which can alleviate the aforementioned problems of Cu/low-k interconnects. Metallic carbon-based interconnects (carbon nano-tubes (CNT), graphene nano-ribbons (GNR)) and optical interconnects are considered two promising alternatives to cope with these problems [118, 109]. CNTs exhibit performance advantages over Cu because the ballistic transport of electrons over micrometer distances results in a much lower resistivity, and strong bonds between carbon atoms create a much higher electromigration tolerance [187]. The performance advantage of CNT over Cu has been confirmed, among others, by G. F. Close et al. [42] through the demonstration of a 1 GHz CNT-integrated oscillator using multiwall (MW) CNT interconnects—which in turn has helped expedite the advent of high performance CNT-based interconnect fabric.

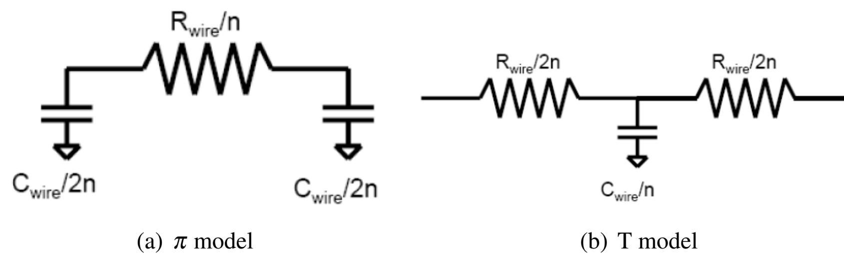
On the other hand, optical interconnects differ fundamentally from the electrical schemes (such as CNT and Cu). First, a large part of the latency and the entire power dissipation is in the end-devices instead of the waveguide. Second, the nature of power dissipation is mostly static rather than dynamic [109, 76]. These differences, when coupled with favorable wire architectures, present new opportunities for optical interconnects. Although promising for most interconnect metrics, the use of optics suffers from the drawback of a relatively large transmission medium (waveguide) pitch ($\sim 0.6\mu m$). The resulting band width density (Φ_{BW}) limitation can be overcome by using the unique wavelength division multiplexing (WDM) option available for optical interconnects [41]. While these new interconnect technologies show promise for meeting future system interconnect requirements, they are currently impractical due to manufacturability limitations, although in theory, they are possible. On the other hand,

a new low swing interconnect circuit scheme—“capacitively driven low-swing interconnect” (CDLSI)—is highly practical, while being equally promising, and hence warrants a detailed analysis [146, 87]. The advantages of CDLSI over the conventional schemes are two-fold: First, it can enormously reduce the energy per bit from a reduced voltage swing. Second, it can achieve a smaller delay. Thus it is the subject of some investigations/research today.

7.4.4 SPICE Model and Performance Metrics

The most important interconnect performance figures are speed, power, signal integrity, and bandwidth. In this subsection, we review a SPICE model for interconnects and specify useful interconnect figures of merit.

SPICE Model: The basic physical properties of metal-based interconnects (Cu or Al) consist of resistance (R), capacitance (C), and inductance (L). In an on-chip interconnect, the inductance value is generally not taken into account because the wire rise (or fall) time is more significant than the time of flight. Therefore, on-chip interconnects can be approximated as a lossy RC network as shown in Figure 7.23.



(a) π model (b) T model

Figure 7.23: One segment of a distributed wire model using SPICE: (a) is a π model and (b) is a T model [87].

In a SPICE simulation, interconnects are modeled as a distributed RC line with n

number of segments. There are two types of lumped electrical wire models. One is the π model and the other is the T model, as shown in Figure 7.23. Ideally, infinite n and infinitesimal segment length are needed for the most accurate delay estimation. However, for the first-order analysis of our SRAMs, the inclusion of either of these models give us the desired accuracy.

Delay Model: Delay is one of the most important performance metrics. Elmore delay is used for a simplified delay calculation of a complex RC network:

$$\tau_{elmore} = \sum_{k=1}^{N} C_k R_{ik}$$
$$R_{ik} = \sum R_j \tag{7.23}$$

The Elmore delay is simply the sum of the RC time constant of each node (between node k and i) with common path resistance (R_{ik}), where k is the index of each node and i is the node where delay need to be measured. Using this simple relationship, the wire delay of the equivalent circuit in Figure 7.24 can be simply given by

$$\tau_{wire} = \alpha \cdot R_{dr}C_p + \beta \cdot R_w C_w + \beta \cdot R_{dr} C_w + \beta \cdot (R_{dr} + R_w) C_L \tag{7.24}$$

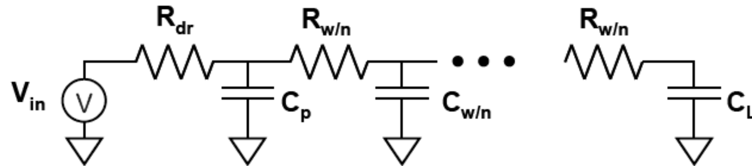


Figure 7.24: Equivalent circuit of a distributed RC interconnect with step input function [87].

where α and β are determined by the type of network and points of interest of the input step response (summarized in Table 7.3). R_{dr} is driver resistance and C_p and C_L are the

Table 7.3: α and β for lumped and distributed networks for different points of interest [87].

Voltage	α (Lumped RC)	β (distributed RC)
0 → 50%	0.69	0.38
0 → 60%	1	0.5
10% → 90%	2.2	0.9
0 → 90%	2.3	1

driver parasitic and transmitter load capacitances, respectively. $R_{w/n}$ and $C_{w/n}$ are the resistance and capacitance of each segment of the wire divided into n segments.

Power Dissipation Model: The power consumption of interconnects can be partitioned into three components: *dynamic*, *static*, and *dynamic short circuit* power. *Dynamic* power dissipation is due to the charging and discharging of the load capacitance (C_L). C_L includes the wire, parasitic, and input capacitance of repeaters. Each time the gate is switched, the charge is either supplied from the power supply to C_L while PMOS transistors dissipate the power or it is drawn to ground with CMOS dissipating the power. Dynamic power is given by

$$P_{dyn} = a \cdot C_L V_{swing} V_{\mathrm{dd}} f_{0 \to 1} \tag{7.25}$$

where a is the switching activity factor and $f_{0\to 1}$ is the frequency of the energy-consuming transitions. *Static* power consumption describes the power dissipation without any switching activity. This includes gate leakage, source-drain leakage, and junction leakage in repeaters.

Putting these all together, static power can be described as

$$P_{stat} = I_{stat} V_{\text{dd}} \tag{7.26}$$

where I_{stat} is the current from V_{dd} to G_{ND} when there is no switching activity. *Dynamic short circuit* power represents the power dissipation due to the current flow when both NMOS and PMOS are in their saturation regions.

Bandwidth/Bandwidth Density: The bandwidth of an interconnect represents its ability to send a certain number of bits per second. If the delay of the interconnect is τ, then ideally the inverse of τ is the number of bits that interconnect can handle within one second. If the interconnect is pipelined or repeated, then the throughput of the interconnect further increases. For example, if the delay of the pipeline segment is τ_{seg}, then the bandwidth of this system is $1/\tau_{seg}$ instead of the inverse of the total wire delay (τ_{seg}). With the advent of multi-core processors, the bandwidth density (Φ_{BW}) has becomes an even more important figure of merit than just the wire bandwidth. This is because a core in the on-chip die tends to have a more limited periphery whereas an interconnect should be laid out within its limit. This makes the chip more bandwidth hungry. Φ_{BW} is given by

$$\Phi_{BW} = \frac{f_{clk}}{W_{pitch}} \tag{7.27}$$

where f_{clk} is the system clock defined by the timing constraints of the system and W_{pitch} is the wire pitch.

Signal Integrity: Noise is one of major concerns in maintaining the correct functionality in a digital system. Noise in digital signaling can be categorized into the noise proportional to signal swing and that which is independent of the signal.

$$V_N = K_N V_S + V_{NI} \tag{7.28}$$

where V_N is noise voltage, V_S is signal voltage, and K_N is a noise coefficient proportional to signal swing. Capacitive coupling cross-talk is the main source of K_N. V_{NI} is a noise source independent of signal swing and is mainly determined by transmitter or receiver offset voltages. Power supply noise is random noise due to the non-ideal impedance of the power supply rail [45]. It is critical to minimize V_N to cope with errors in digital signaling.

7.4.5 Existing and Future Interconnects

This subsection begins with a brief description of the basic physical properties of different materials that make up the structure of the *local*, *semi-global*, and *global* interconnects used in existing and future circuits. This subsection concludes with a few plots illustrating a performance comparison between the various interconnects.

Cu

As wire cross-sectional dimension and grain size become comparable to the bulk mean free path (mfp) of electrons in Cu, two major physical phenomena occur to the current-carrying electrons and bulk phonons as illustrated in Figure 7.25(a). One is interface scattering and the other is grain boundary scattering [87]. These increase Cu resistivity above the ideal

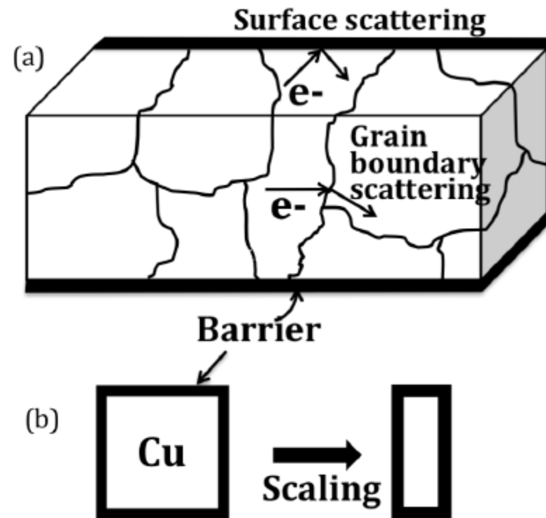


Figure 7.25: (a) Schematic illustration of the surface and grain boundary scatterings, and the barrier effect. (b) Impact of scaling on barrier effect. Cu can be scaled while barrier cannot [87].

bulk resistivity ($\rho_0 = \Omega \cdot cm$) [75]. Based on the model presented by Kyung-Hoae Koo [87], for the 22-nm node, Cu resistivity for the minimum width wire increases to $5.8\mu\Omega \cdot cm$ ($\sim 3\times$ that of bulk).

On the other hand, Cu interconnects typically need a diffusion barrier, which can come in the form of Tantalum (Ta), Ruthenium (Ru), and Magnesium (Mg) based materials. Because the resistivity of these materials is much higher than that of Cu, in effect, the barrier reduces the useful interconnect cross-section. One way to capture this effect is to define the problem in terms of an increased effective resistivity, which would be applicable to the original cross-sectional area. Figure 7.25(b) shows that the down-scaling exacerbates the barrier problem since the barrier thickness does not scale proportionately to the cross-sectional interconnect scaling. This, in turn, results in an increase in the effective resistivity with smaller technology nodes. Three dashed curves in Figure 7.26 quantify this effect. The effective resistivity here also

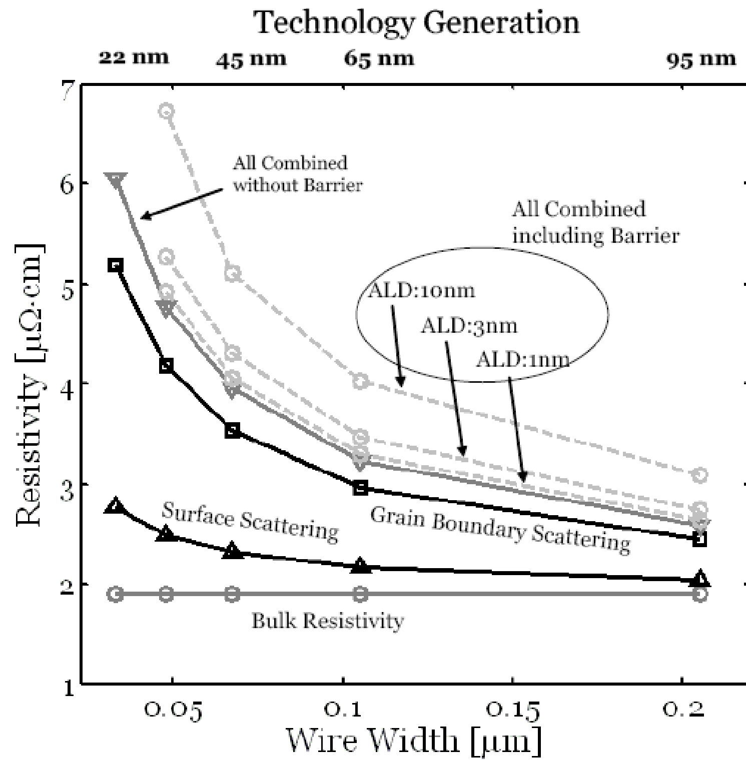


Figure 7.26: Cu resistivity in terms of wire width taking into account the surface and grain boundary scattering and barrier effect. The barrier layer is assumed to be uniformly deposited, e.g., using atomic layer deposition (ALD) [87].

captures both the surface and grain boundary scatterings. It is clear that with technology scaling, effective resistivity increases dramatically. Further, lowering the barrier thickness (ALD: 3nm vs. 1nm) has a big impact on effective resistivity [75].

The increase in wire length (l), in addition to the reduction in cross-sectional area (A), further increases wire resistance, subsequently limiting the signal rise time and the bandwidth. This can be well understood from the simple relationship between the ideal bit rate (B) and the cross-sectional area and the wire length, shown in Figure 7.27. Typically, buffering the wire with multiple repeaters mitigates the bandwidth shortfall. However, it consumes a significant portion of the power budget. Thus, for the electrical interconnect, it becomes more difficult to

meet the bandwidth requirement and power budget simultaneously [109].

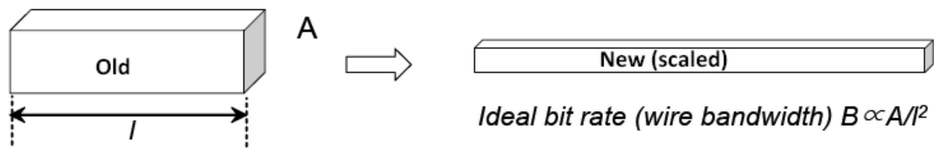


Figure 7.27: The impact of interconnect scaling. Scaled wire with lower A and longer l has higher resistance resulting in higher delay, increased power, and reduced bandwidth) [87].

CNT

Carbon nano-tubes (CNTs) have attracted great attention because of their interesting physical and electrical properties [187]. Their near one-dimensional shape supports ballistic transport, making them potentially useful in many applications, such as transistors, sensors, and interconnects. In addition, CNTs offer great mechanical strength due to their strong sigma (σ) bonds between neighboring carbon atoms. These excellent physical properties have been proven theoretically and experimentally by intensive research for more than a decade. CNTs can be categorized as *semiconducting* or *metallic* depending on their chiral configurations. Used as an interconnect, typically *metallic* carbon nano-tubes are considered. Depending on the shape, a CNT can be categorized as a *single-walled carbon nano-tube* (SWCNT) or a *multi-walled carbon nano-tube* (MWCNT). A SWCNT is constructed by wrapping a graphene layer into a cylindrical shape. Its diameter ranges from 0.4nm to 4nm, but is typically on the order of 1nm. A MWCNT is formed by multiple layers of SWCNTs with different diameters. Its diameter ranges from 10nm to 100nm. Figure 7.28 shows the difference between SWCNT and MWCNT.

The resistance of a CNT bundle depends on the total cross-sectional area of the wire and the fractional packing density (*PD*) of metallic CNTs within it [117]. For a SWCNT, in

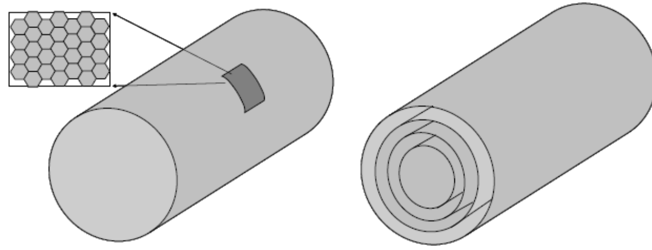


Figure 7.28: Three dimensional illustration of (a) SWCNT, (b) MWCNT [87].

the absence of any type of scattering, the maximum quantum conductance is given by Equation (7.29).

$$G_{SWCNT} = \frac{4q^2}{h} = \frac{2q^2}{\pi\eta} \tag{7.29}$$

where h is the Planck constant and q is the charge of one electron. The multiplier of four accounts for the two channels due to electron spin and another two channels due to sub-lattice degeneracy. Thus, the quantum resistance of a SWCNT is 6.45 $K\Omega$. This resistance is generally too large to allow its use as an interconnect. One way to get around this problem is to use a bundle of SWCNTs.

It is very important to investigate the transmission line component of a SWCNT to understand its applicability as an interconnect. Figure 7.29 shows a transmission line *LC equivalent* circuit of a SWCNT.

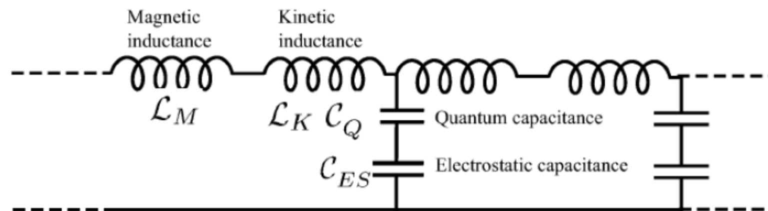


Figure 7.29: Transmission line LC components of SWCNT [87].

The calculation and analysis of the delay of the SWCNT transmission line is beyond the scope of this thesis. The interested reader may consult the references used throughout this subsection. Instead, we present a comparison of the important characteristics of CNT and Cu interconnects (Figure 7.30(a)). In addition, we compare the L/R ratio of a CNT to that of Cu (Fig. 7.30(b)), and we show a simulated step response of a repeated CNT wire along with that of Cu (Figure 7.30(b) inset).

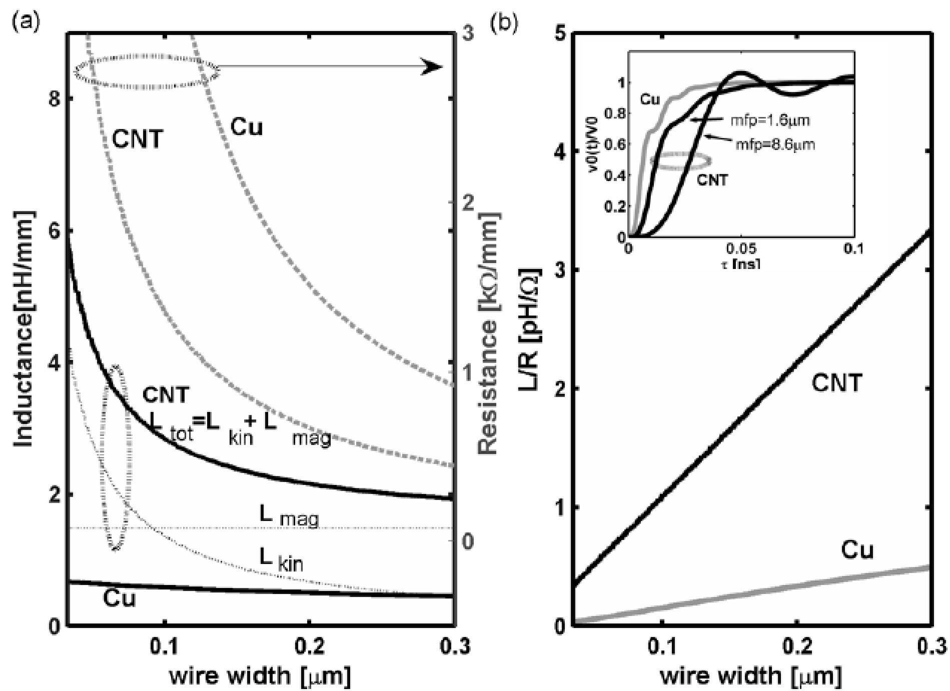


Figure 7.30: (a) Inductance and resistance of Cu and CNT vs. wire width. Wire width is in *global* interconnect regime. Fractional packing density (*PD*) is assumed to be 33%. l_0 (the electron mean free path) is 1.6μm. L_{tot} (total inductance) consists of combination of L_{kin} (kinetic inductance) and L_{mag} (magnetic inductance) (b) Inductance to resistance ratio as a function of the wire width. (b. inset) step response of Cu and CNT wire with optimized spacing (*h*). Wider interconnect implies more ripples in time domain response due to higher transmission line effect [87].

Figure 7.30(a) illustrates the plot of total inductance (L_{tot}) of a CNT bundle along with its components. Smaller widths render a larger L_{tot} because of an increase in kinetic in-

ductance (L_{kin}). Cu L_{tot} is lower than CNT L_{tot} for all widths, but Cu resistance is higher than that of CNT bundle resistance due to Cu's shorter mean free path (mfp), (see Figure 7.30(a)). As seen in Fig. 7.30(b), the above results translate into a $6\times$ larger inductance to resistance ratio (L/R) for a CNT-bundle as compared to Cu wires, indicating a more pronounced impact of inductance for the CNT bundle. A simulated step response of a repeated CNT wire shows a significantly under-damped frequency response as compared to Cu (Figure 7.30(b) inset), confirming the importance of inductance in CNTs. Figure 7.30(b) also shows that the L/R ratio is higher for larger widths. Thus, a full RLC model is imperative for CNT bundles especially for wider *global* wires. Inductance can be ignored for thinner *local* wires.

GNR

A graphene sheet is an ideal two-dimensional carbon honeycomb structure. Graphene nano-ribbons, abbreviated as GNRs, are edge-terminated graphene sheets, as shown in Figure 7.31.

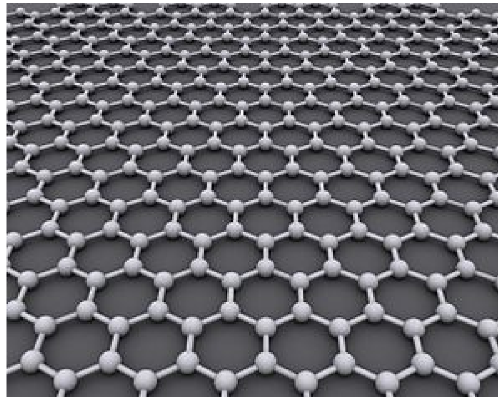


Figure 7.31: Graphical illustration of 2-D Graphene nano-ribbon (GNR) [56].

They are equivalent to unrolled SWCNTs. GNRs share most of their physical and electrical characteristics with CNTs, such as becoming semiconducting or metallic depending

on the chirality of their edges [2]. However, GNRs can be fabricated in a more controllable process, such as optical lithography, while CNT growth results in a random chiral distribution. Thus, GNRs could be a very good interconnect material.

In Figure 7.32, it is shown that a *metallic* GNR cannot outperform a Cu interconnect until the width is less than 7nm. This is due to diffusive edge scattering, which significantly limits the performance of the GNR interconnect. In addition, the resistance of a GNR is higher than that of the mono-layer SWCNT for all ranges of wire width. However, if we use a multilayered GNR interconnect, it can improve the performance [87].

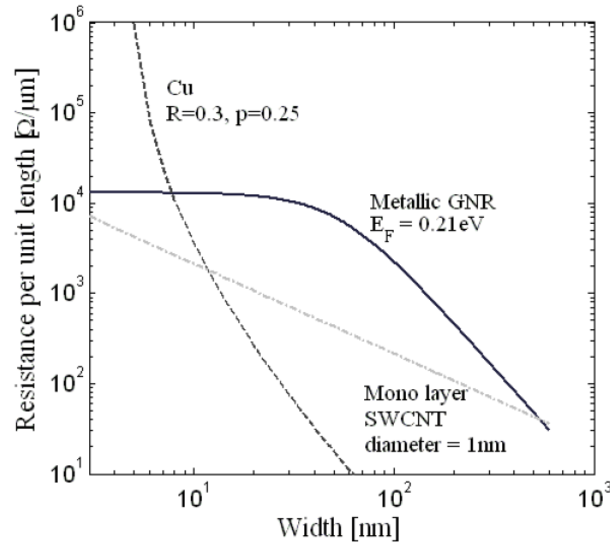


Figure 7.32: Resistance comparison between GNR, mono-layer SWCNT, and Cu. The Fermi-level is assumed to be 0.21eV, as reported in an experimental result [2].

Repeaters

Figure 7.33(a) is the schematic representation of a buffered interconnect. If we consider an interconnect of total length *L*, it is buffered at length *l* to achieve minimized delay. The length *l*, along with a methodology for optimally buffering the interconnect, is outlined by

Koo [87].

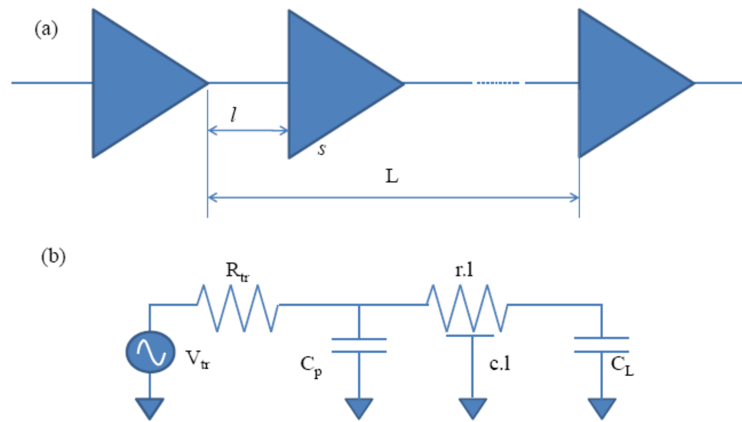


Figure 7.33: (a) Schematic of an optimally buffered interconnect. The total length is L and l is the optimal distance between repeaters to minimize delay. s refers to the optimal size of the input transistor. Each repeater has a fanout of one (FO). (b) The equivalent circuit of one segment with l and s [87].

Figure 7.33(b) shows a distributed RC network, indicating one segment of the repeated interconnect with length l. s refers to the optimal size of the input transistor. V_{tr} is the voltage source at the input stage. R_{tr} is the driver resistance which has dependence on the transistor size, C_p is the output parasitic capacitance of the driver, and C_L is the load capacitance of the receiving end. r and c are the interconnect resistance and capacitance per unit length, respectively. The detailed computation of the segment delay of this interconnect, τ_0, can be found in Kaustav Banerjee et al. paper [14]. The performance study and the subsequent circuit model of a repeater are illustrated at the end of this subsection.

Koo [87] suggests using RLC models for global/semiglobal Cu and CNT wires and he includes repeater insertion for delay reduction (Figure 7.34). The values of R, C, and L are discussed by Kaustav Banerjee et al. [14].

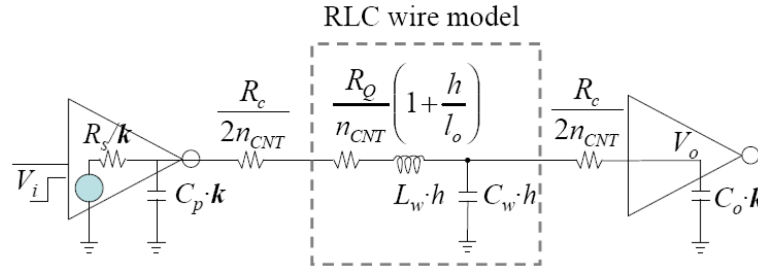


Figure 7.34: Equivalent circuit model of a repeater segment for CNTs. k is the optimized repeater area. h is the optimized wire length per repeater. R_s is inverter output resistance. R_c and R_Q are the contact resistance and quantum resistance of SWCNT, respectively. L_w and C_w are the inductance and the capacitance of CNT bundle, respectively. C_p and C_0 are the input capacitance and the output parasitic capacitance of the inverter, respectively. n is the number of segments the length L is divided into [87].

Optical

The basic architecture of an optical link consists of an off-chip laser, a quantum-well modulator at the transmitter (converts the CMOS gate output to an optical signal), a waveguide comprising a silicon core (refractive index$\sim$3.5), SiO2 cladding as the transmission medium, and a transimpedance amplifier (TIA) followed by gain stages at the receiver (Figure 7.35).

The total delay of an optical interconnect is the sum of the transmitter, waveguide, and the receiver delays. The transmitter delay arises from the CMOS gate driving the capacitive modulator load. It is minimized using a buffer chain and is dependent on the fan-out-four (*FO4*) inverter delay of a particular technology node. The waveguide delay is dictated by the speed of light in a dielectric waveguide ($\sim$11.3ps/mm). Finally, the receiver delay is calculated based on circuit considerations. It assumes the input pole (the node at the input of TIA) to be dominant and is optimized to meet the bandwidth and the bit error rate (10^{-15}) criteria. The total power dissipation for the optical interconnect is calculated by optimizing the sum of the receiver and transmitter power dissipation as outlined by P. Kapur et al. [76] in the context of off-chip interconnects. For the performance study plots shown at the end of this subsection,

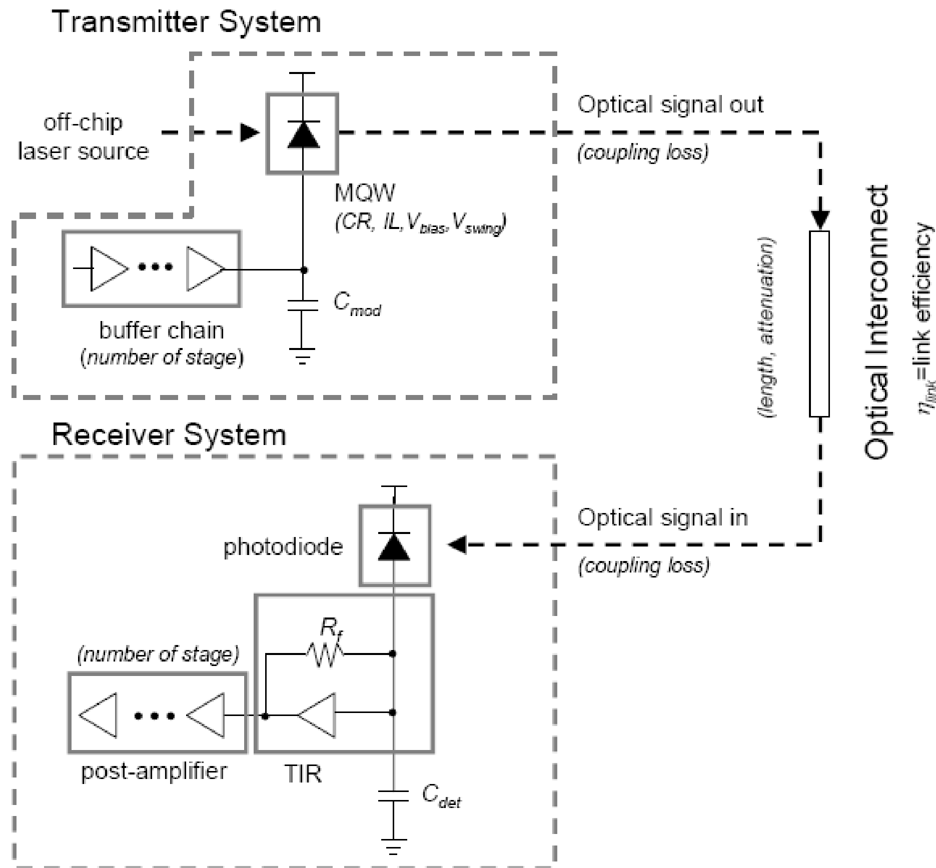


Figure 7.35: The schematic of a quantum-well modulator-based optical interconnect. The modulator parameters assumed for this optical interconnect were taken from J. K. White et al. [83]. MQW stands for "multiple quantum-well." C_{mod} is the capacitance of the MQW modulator. C_{det} is the capacitance of the detector.

the following values were assumed: insertion loss (*IL*)=0.475, contrast ratio (*CR*)=4.6, and bias voltage (V_{bias})=4.7V.

7.4.6 Performance comparison between Cu/low-k, m-SWCNT Bundle, and Optical Interconnects

Figure 7.36 compares the interconnect latency of CNT-, Cu-, and optics-based links as a function of the technology node for *semiglobal* (∼1mm) and *global* ((∼10mm) wires. For

mean free paths (l_0) of 0.9μm and 2.8μm, CNT wires show 1.6$\times$ and 3$\times$ latency improvement, respectively, over Cu at all technology nodes. Optical wires show an advantage over both CNT and Cu for longer lengths ($\sim$10mm) because a large fraction of the delay occurs in end-devices.

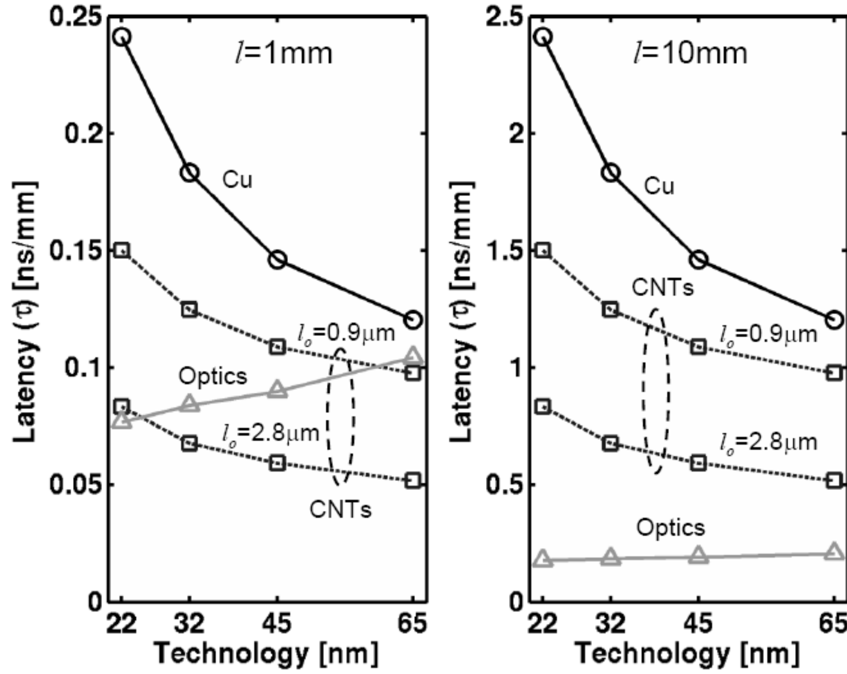


Figure 7.36: Latency as a function of technology node for two different interconnect lengths. l_0 is the mean free path and *PD* is packing density of metallic SWCNTs in a bundle. SWCNT diameter (*dt*) is 1nm. For optics, the capacitance of monolithically integrated modulator/detector (C_{det}) is 10fF [125, 50].

For 1mm long wires, optical interconnects become advantageous over CNT only at smaller technology nodes. This is because, with scaling, CNT and Cu latency increases, whereas optical delay latency decreases due to an improvement in transistor performance (transmitter and receiver).

Figure 7.37 compares the energy per bit requirements of the three technologies. For Cu/CNT wires the dominant energy is the dynamic switching energy: CV^2 (C is total capacitance that includes the wire and the repeater components, V is the voltage). For optical in-

terconnects, the static power dissipation in the end-device amplifiers dominates. For 10mm *global* wires, optics is most energy efficient, while for 1mm *semiglobal* wires, both Cu and CNT present a better efficiency than optics.

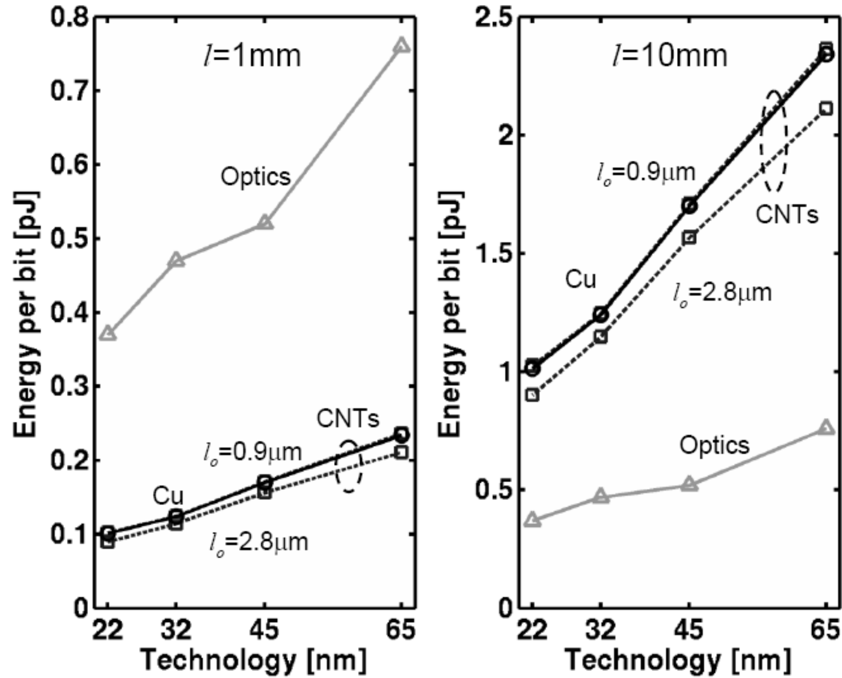


Figure 7.37: Energy per bit vs. technology node for two different interconnect lengths corresponding to *global* and *semiglobal* wire length scales. For CNTs, *PD* is 33% and the wire diameter *dt* is 1nm. For optics, the capacitance of a monolithically integrated modulator/detector (C_{det}) is 10fF [125, 50].

From Figure 7.37, we can also observe that, at all length scales and at the 22-nm node, CNTs with l_0=2.8μm are 20% more energy efficient as compared to Cu. This is because a CNT operates in the RLC region, where a smaller resistance results in a smaller optimum repeater size; hence a smaller total repeater capacitance [68]. This is in contrast with an RC wire, where the total optimum repeater capacitance is a constant fraction of the wire capacitance, irrespective of the resistance. The 0.9μm l_0 CNT exhibits similar energy per bit as Cu because even though l_0 is larger than Cu, the sparse (33%) *PD* results in a resistance similar to that of Cu.

Figure 7.38(a) explores the dependence of latency on the wire length between Cu/CNTs and optics. The delay in all three technologies linearly increases because Cu/CNTs are buffered with repeaters in a delay-optimized fashion, and for optics, the latency of the medium is linearly dependent on the length.

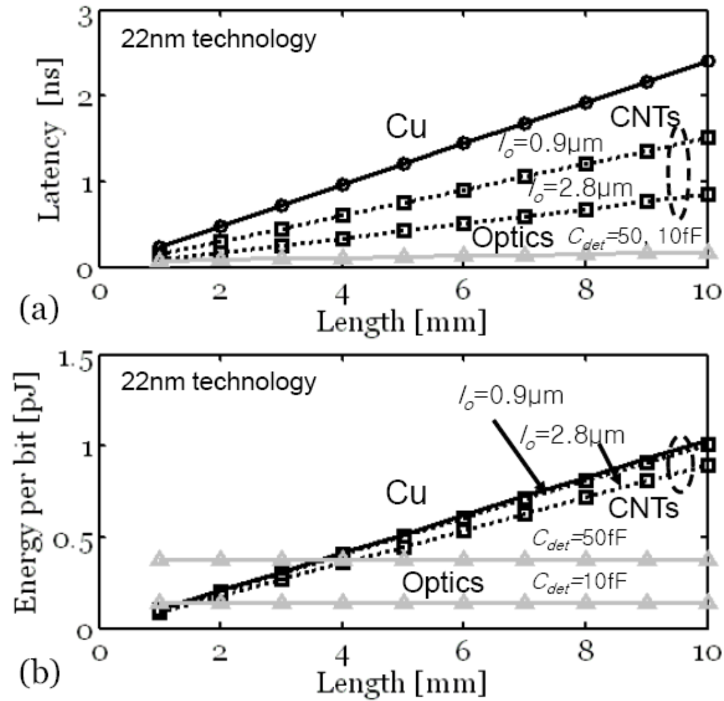


Figure 7.38: Latency and energy per bit in terms of wire length for the 22-nm technology node. l_0 is the mean free path (MFP). For optics, the detector capacitance (C_{det}) is 50fF and 10fF [87].

Optics clearly shows the lowest latency for the reasons stated earlier in this subsection. CNTs with l_0=0.9μm and l_0=2.8μm give 1.6× and 3× latency improvement over Cu, respectively, as shown in Figure 7.38(a).

Figure 7.38(b) displays the energy per bit as a function of the wire length of the three technologies. While Cu/CNT interconnects give a linearly increasing energy per bit as length increases, optical interconnects show almost constant values at all length scales. This is because

the static power from the end devices in optics does not scale with length, whereas dynamic power in electrical wires does. Here, below the 4mm (*semi-global* interconnect), CNTs with l_0=2.8μm can outperform the optics. In addition, below the length of 3.8mm, Cu and CNTs with l_0=0.9μm can outperform the optics.

The performance of CNT is a strong function of l_0 and *PD*, whereas optical wire performance critically depends on capacitances associated with the modulator and the detector (C_{mod}, C_{det}). The impact of these device and materials parameters is quantified in the following two plots. Such an analysis is useful for getting an idea of the required parameters from a system standpoint.

Figure 7.39 illustrates that an improvement in CNT l_0 from 0.9μm (practical) to 2.8μm (ideal) and in *PD* from 0.33 to 1 results in reduced power density for all Φ_{BW}. Moreover, the improvement in l_0 has a larger impact than an improvement in *PD*. This is because a *PD* increase results in a smaller increase to the L/R ratio since L_{kin} also decreases along with *R*. For this *SA* and for C_{det}=25fF, CNT and Cu outperform optics. However, if the C_{mod} and C_{det} can be reduced to 10fF using a monolithic detector (as opposed to hybrid bonded III-V detector), optical wires outperform other technologies even at smaller *SA*.

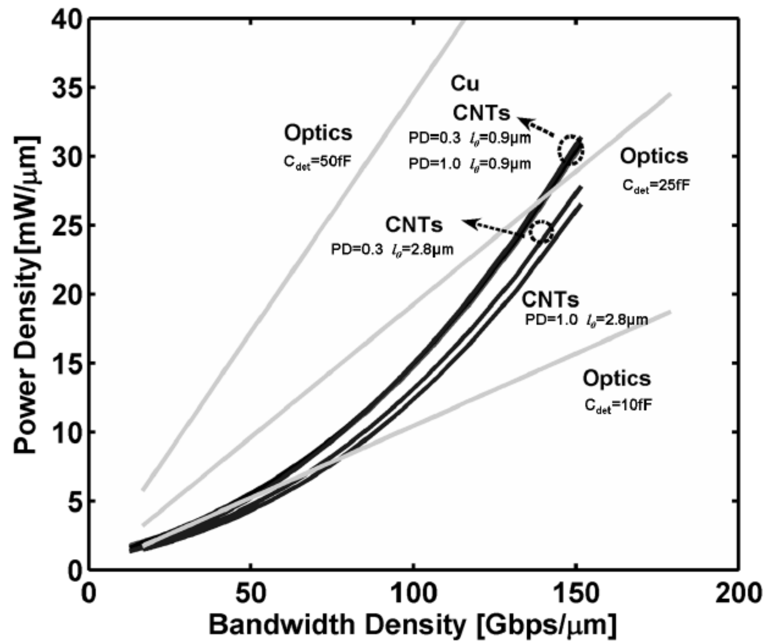


Figure 7.39: The impact of CNT and optics technology improvements on power density vs. bandwidth density (SA = 20%). C_{det} reduction for optics results in a large improvement in power density. (wire length = 10mm, 22-nm transistor technology node, f_{clk}=10Gb/s) [87].

Similarly, Figure 7.40 captures the impact of technology parameters on the latency. For CNTs, improvement in both l_0 and PD results in a substantial decrease in latency compared to Cu. With ideal l_0 and PD, and at lower Φ_{BW}, latency performance is comparable to optical wires.

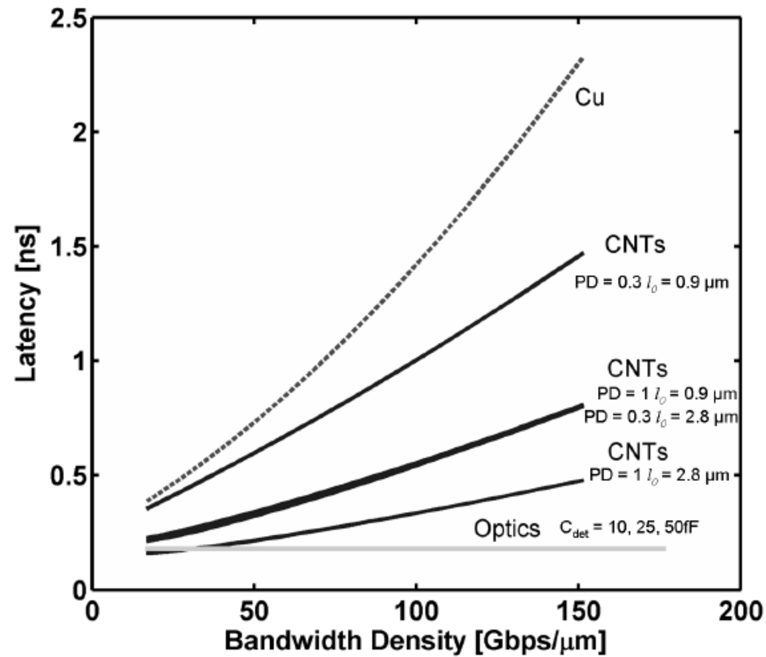


Figure 7.40: The impact of CNT and optics technology improvement on latency vs. bandwidth density. CNT parameter improvement results in a very large improvement in latency over Cu. A CNT with ideal parameters has a comparable latency to optical wires at very low Φ_{BW}. (Wire length = 10mm, 22-nm transistor technology node, f_{clk}=10Gb/s) [87].

7.4.7 Capacitively Driven Low-Swing Interconnect (CDLSI)

The most common power reduction technique is to reduce voltage swing and decrease dynamic power dissipation from capacitive wires. Therefore, to reduce dynamic power, newer circuits use conventional low-swing interconnects that transmit the signals (generated by logics operating at normal voltage) at a reduced voltage. However, such power savings are typically accompanied by a latency penalty and a reduction in the noise margin. Moreover, conventional low-swing interconnects usually require a secondary low voltage power supply, which makes the system more expensive and complex [194].

Figure 7.41 shows the basic concept of a low-swing interconnect with a driver and a

receiver. An incoming signal with full rail-to-rail swing is converted to reduced swing through a driver (usually a level shifter with a different supply voltage). Then, the reduced swing propagates through the diffusive RC wire. Finally, a receiver regenerates the low-swing signal to a full-swing signal via another level shifter. Figure 7.42 illustrates a more detailed view of such a system using two level shifters (one on either sides of C_L) [149].

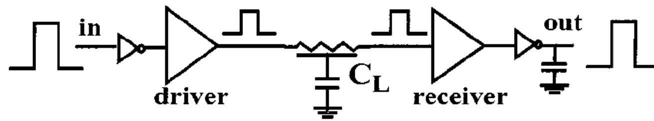


Figure 7.41: Schematic of conventional low-swing interconnect scheme [141].

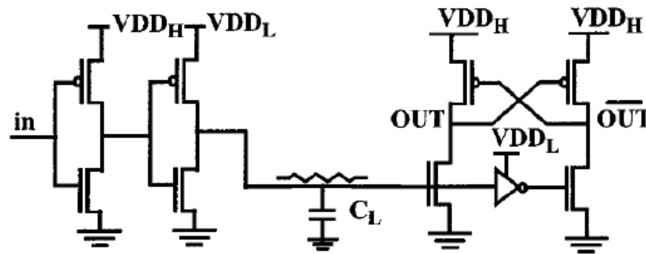


Figure 7.42: Conventional low-swing scheme with additional power supply. VDDH is the normal/high rail-to-rail supply voltage. VDDL is the low supply voltage [141].

Recently, a new Capacitively Driven Low-Swing Interconnect (CDLSI) was shown to exhibit excellent energy savings without seriously impacting latency (Figure 7.43) [64, 108]. The key element in this system is the coupling capacitance, which not only eliminates the necessity of a secondary power supply, but also introduces preemphasis (the high frequency emphasis arising from the pole-zero pair of the high pass filter network), resulting in bandwidth improvement. Wires are differential and twisted in order to cancel the coupling noise. The coupling capacitor (C_c) is inserted between the transmitter and the wire. The receiver is comprised of a strong arm latch sense amplifier followed by an RS latch (Figure 7.43). The sense amplifier

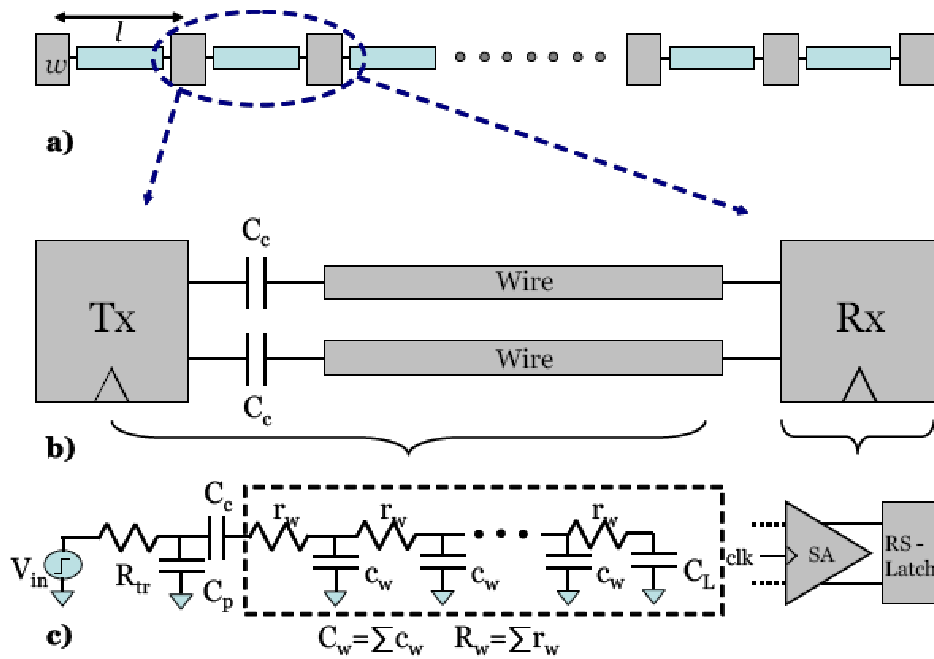


Figure 7.43: (a) Simple illustration of repeated capacitively driven low-swing interconnect (CDLSI). l is a segment length. w is the size of Tx/Rx. (b) Zoomed schematic of one segment of CDLSI. C_c is coupling capacitor. (c) Equivalent circuit model of one segment. The dashed box indicates a distribution model of the wire. R_{tr} is an effective resistance of a NAND gate when pull up and pull down resistances are equalized. C_p is output capacitance of the transmitter. R_w and C_w are wire resistance and capacitance, respectively. C_L is load capacitance corresponding to the input of the sense amplifier. Fanout between the sense amplifier (SA) and the RS latch is assumed to be 1 ($w_{SA} = w_{RS} = w$) [87].

boosts a reduced swing to full rail-to-rail swing. The common mode of this amplifier is set at the supply voltage through a PMOS transistor on the receiver side. The RS latch is inserted in order to register the output value of the sense amplifier.

7.4.8 Performance comparison between CDLSI, Cu/low-k, CNT, and Optical Interconnects

Figure 7.44 compares the lowest delay as a function of required Φ_{BW}. CDLSI shows better a delay performance over the conventional wire only in the low Φ_{BW} domain. According

to Koo [87], the Φ_{BW} window of CDLSI is limited by two factors: first, its low intrinsic wire bandwidth caused by high transceiver delay and second, the 2× wire pitch occupancy of CDLSI due to differential signaling. CDLSI's delay advantage over the conventional wire degrades as the system requires higher Φ_{BW}. This is because when the wire becomes too small (indicated by higher Φ_{BW}), it results in a much too high resistivity—which demands an excessive number of transceivers, canceling the advantage of the pre-emphasis effect.

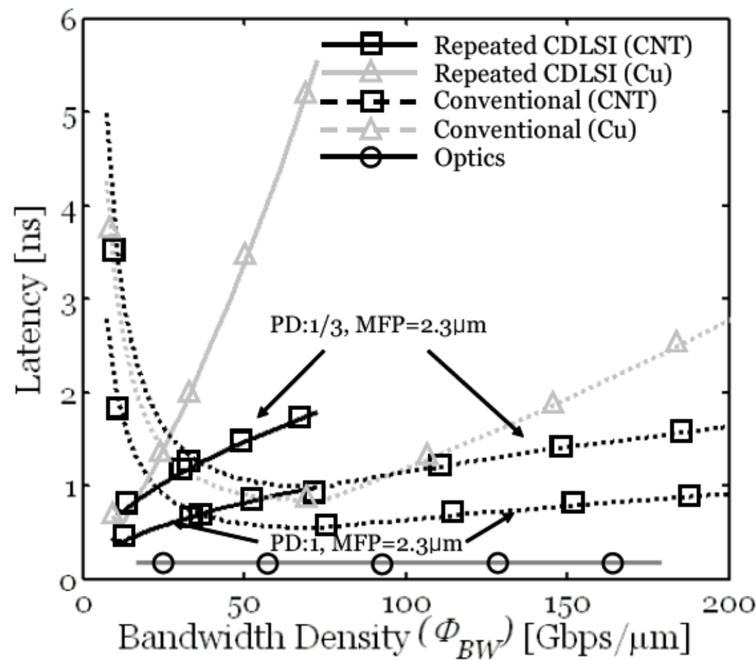


Figure 7.44: Delay vs. bisectional bandwidth density (Φ_{BW}). For optics, WDM technique is assumed. (Total 10 wavelengths with 10Gbps/ch bandwidth) [87].

CNTs with a higher filling factor (*PD*=1) can reduce delay further. For optics, a larger Φ_{BW} is achieved using denser wavelength division multiplexing (WDM) and the maximum Φ_{BW} is limited by the degree (number of wavelengths) of WDM [88]. Optics shows the smallest delay as compared to Cu and CNT technologies. Finally, Figure 7.45 compares energy density

($pJ/\mu m$) as a function of system required Φ_{BW}. Here, CDLSI also loses its energy advantage over conventional wires for high Φ_{BW} for the same reason as in Figure 7.44. Optics, although difficult to inexpensively implement, shows the best performance.

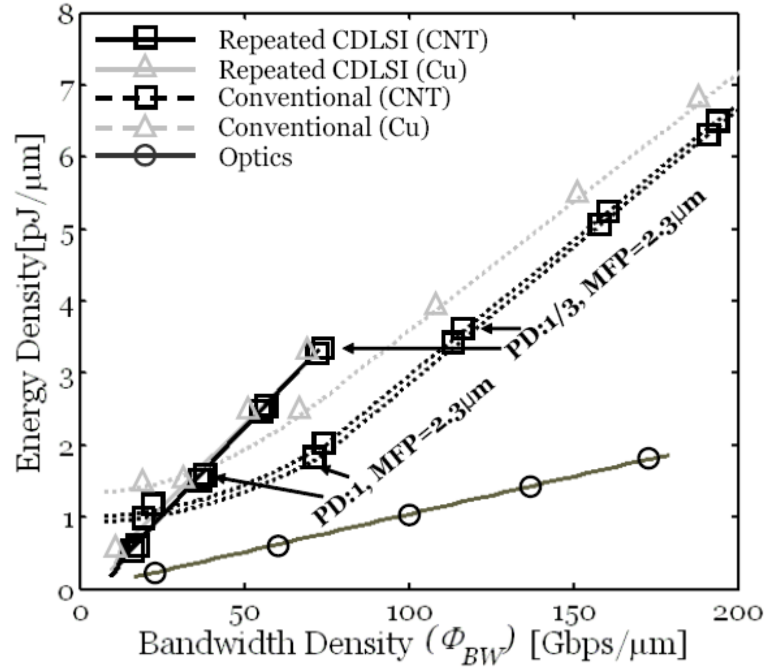


Figure 7.45: Energy Density vs. bisectional bandwidth density (Φ_{BW}). For optics, WDM technique is assumed. (Total 10 wavelengths with 10Gbps/ch bandwidth) [87].

7.5 Major Techniques for Leakage Control in Caches/SRAMs

This section discusses three of the major techniques commonly used to keep the transistor leakage current in caches and/or SRAMs in check. There have been several proposed models (ORION 2.0, Hotleakage, SimpleScalar/Wattch, etc.) during the last decade, each of which uses its own slightly different method to achieve the goal of controlling/minimizing the leakage current.

7.5.1 Lowering the Quiescent Vdd (Gated-Vss)

Leakage currents decrease as the supply voltage (V_{dd}) is lowered. The *gated-*V_{dd} structure was introduced by M. Powell et al. [137] as a way to reduce leakage power by using a high threshold "header" transistor to disconnect a cell, row, or way in the cache from V_{dd} [195]. This high-threshold transistor drastically reduces the leakage of the circuit because the high-threshold transistor effectively breaks the connection to the power supply. While this technique is very efficient in preventing leakage, there is the disadvantage that the cell loses its state (information). Thus this is called a *state-losing technique*. This means that there will be a performance penalty when the data in the cell is accessed and needs to be fetched from a deeper level of the cache. This technique was used by S. Kaxiras et al. [79] to shut down lines in a cache to prevent leakage. Because the sleep transistor is more effective as a "footer" on the connection to ground—it is easier to prevent bitline leakage this way—this technique is better called *gated-Vss*.

7.5.2 Multiple Threshold CMOS (MTCMOS)

It is clear from the above discussion that the threshold voltage is one of the most important parameters influencing the leakage current. The multiple threshold CMOS technique was proposed by K. Nii et al. [123]. For the active mode, the low threshold voltage is preferred because of the high performance. However, for the standby mode of operation, the high threshold voltage is useful for the reduction of the leakage power. Hence, if the transistors can be set to different threshold voltages, most likely using reverse-body-bias (RBB), then the threshold voltage can be set according to the different modes of operation. This does not lose the data

stored in a cell, so this is a *state-preserving technique*. There is still some overhead, however, when accessing a unit that is in standby mode because the threshold voltage must be returned to the proper level before the value can be read [195].

7.5.3 Drowsy Caches

This method, proposed by K. Flautner et al. [53], utilizes dynamic voltage scaling to reduce the supply voltage of the cell to approximately 1.5 times V_{th}. This reduces leakage current dramatically due to short-channel effects and preserves the value that is stored, making this another *state-preserving technique*. Like MTCMOS, there is still some overhead because V_{dd} must be returned to the proper level before the value can be read.

For all of the above techniques, during the initialization phase of the simulation, the leakage currents for the cache/SRAM (with and without the specified technique turned on) are calculated using one of the available models. Subsequently, the leakage energy of the cache/SRAM is calculated for every cycle using the calculated leakage current and the turn-off ratio (the fraction of cache/SRAM when using one of the above leakage saving techniques to when not using any leakage saving technique).

7.6 Power, Leakage, and Energy Delay

7.6.1 Power Overview

We have seen that the use of a 6T-cell—with its symmetrical shape, reliable logic swing, and sufficiently high noise margins—offers a superior robustness, which simplifies the

design process considerably and opens the door for SRAM design automation. Another advantage of 6T-cell (using static CMOS technology) is the tolerable power consumption in the steady-state operation mode, when clever designs keeping the leakage current in bound are used. It is this combination of robustness and low static power that has made 6T-cell the choice of most contemporary SRAM/cache designs, including ours.

Considering that the core of each 6T-cell is composed of two back-to-back inverters (Figure 4.1), we look at the total power consumption of the CMOS inverter. It can be expressed as the sum of its three components:

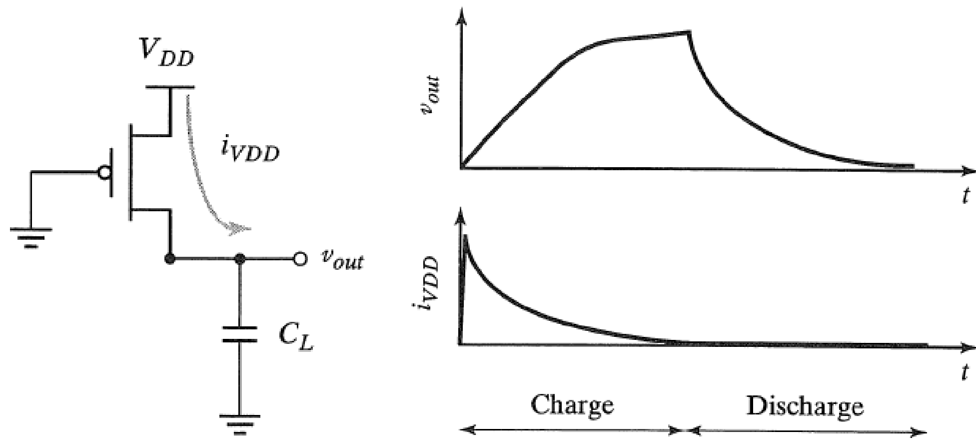
$$P_{tot} = P_{dyn} + P_{dp} + P_{stat} = (C_L V_{\mathrm{dd}}^2 + V_{\mathrm{dd}} I_{peak} t_s) f_{0\rightarrow 1} + V_{\mathrm{dd}} I_{leak} \tag{7.30}$$

The power dissipation of SRAM (using CMOS circuits) is by far dominated by the dynamic dissipation (P_{dyn}) that results from the charging and discharging capacitances. The direct-path consumption (P_{dp}) can be kept within bounds by careful design, and thus should not be an issue. Unlike in older nodes (i.e. before 45-nm), the leakage of recent and future nodes (P_{stat}) is not ignorable, particularly due to it exponential dependency on the transistor threshold voltage, It must be kept in check by using architectural techniques, such as putting the non-active sections of the SRAM circuit into sleep mode. The next three sub-sections discuss the three components of Equation (7.30) in more detail.

7.6.2 Dynamic Power Consumption

Dynamic Dissipation due to Charging and Discharging Capacitances:

In a very simplified circuit (Figure 7.46), each time the capacitor C_L gets charged through the PMOS transistor, its voltage rises from 0 to V_{dd} and a certain amount of energy is drawn from the power supply. Part of this energy is dissipated in the PMOS device, while the remainder is stored on the load capacitor. During the high-to-low transition, this capacitor is discharged, and the stored energy is dissipated in the NMOS transistor.



(a) Equivalent circuit during the low-to-high transition (b) Output voltages and supply current during (dis)charge of C_L

Figure 7.46: Dynamic Dissipation due to Charging and Discharging Capacitances [141].

A precise measure for this energy consumption can be derived. Let us first consider the low-to-high transition. We assume, initially, that the input waveform has zero rise and fall times—in other words, the NMOS and PMOS devices are never on simultaneously. Therefore, the equivalent circuit of Figure 7.46(a) is valid. The corresponding waveforms of $v_{out}(t)$ and $i_{\mathrm{Vdd}}(t)$ are pictured in Figure 7.46(b).

The values of the energy E_{Vdd} taken from the supply during the transition, as well as the energy E_C stored on the capacitor at the end of the transition, can be derived by integrating

the instantaneous power over the period of interest [141].

$$E_{\text{Vdd}} = \int_0^{\infty} i_{\text{Vdd}}(t) V_{\text{dd}} dt = V_{\text{dd}} \int_0^{\infty} C_L \frac{dv_{out}}{dt} dt = C_L V_{\text{dd}} \int_0^{V_{\text{dd}}} dv_{out} = C_L V_{\text{dd}}^2 \tag{7.31}$$

$$E_C = \int_0^{\infty} i_{\text{Vdd}}(t) v_{out} dt = \int_0^{\infty} C_L \frac{dv_{out}}{dt} v_{out} dt = C_L \int_0^{V_{\text{dd}}} v_{out} dv_{out} = \frac{C_L V_{\text{dd}}^2}{2} \tag{7.32}$$

These results can also be derived by observing that, during the low-to-high transition, C_L is loaded with a charge $C_L V_{\text{dd}}$. This charge requires an energy from the supply equal to $E_{\text{Vdd}} = C_L V_{\text{dd}}^2 = Q \times V_{\text{dd}}$. The energy stored on the capacitor equals $C_L V_{\text{dd}}^2/2$. This means that only half of the energy supplied by the power source is stored on C_L. The other half has been dissipated by the PMOS transistor. Notice that this energy dissipation is independent of the size (and hence the resistance) of the PMOS device! During the discharge phase, the charge is removed from the capacitor, and its energy is dissipated in the NMOS device. Once again, there is no dependence on the size of the device. In summary, each switching cycle (consisting of an L→H and an H→L transition) takes a fixed amount of energy, equal to $C_L V_{\text{dd}}^2$. In order to compute the power consumption, we have to take into account how often the device is switched. If the gate is switched on and off $f_{0\to 1}$ times per second, the power consumption is given by

$$P_{dyn} = C_L V_{\text{dd}}^2 f_{0\to 1} \tag{7.33}$$

where $f_{0\to 1}$ represents the frequency of the energy-consuming transitions (these are 0→1 transitions for static CMOS).

Advances in technology result in ever-higher values of $f_{0\to1}$ (as t_p decreases). At the same time, the total capacitance of the chip (C_L) increases as more and more gates are placed on a single die. Consider, for instance, a 16-nm CMOS chip with a clock rate of 5 GHz, an average load capacitance of 1.5 fF/gate, and assuming a fan-out of 4. The power consumption per gate for a 0.9V supply then equals approximately 60μW. For a design with 10 million gates, and assuming that a transition occurs at every clock edge, this would result in a power consumption of 60W! This evaluation, fortunately, presents a pessimistic perspective. In reality, the actual activity in the circuit is substantially lower because not all gates switch at the full rate of 5 GHz, and even if some of them did, the output does not swing from rail-to-rail. The power dissipation will thus be substantially lower.

Computing the power dissipation of a circuit is complicated by the $f_{0\to1}$ factor, also called the switching activity. While the switching activity is easily computed for an inverter, it turns out to be far more complex in the case of gates and circuits. One concern is that the switching activity of a network is a function of the nature and statistics of the input signals: If the input signals remain unchanged, no switching occurs, and the dynamic power consumption is zero! On the other hand, rapidly changing signals provoke plenty of switching and therefore power dissipation. Other factors influencing the activity are the overall network topology and the function to be implemented. We can accommodate this by writing

$$P_{dyn} = C_L V_{\text{dd}}^2 f_{0\to1} = C_L V_{\text{dd}}^2 P_{0\to1} f \tag{7.34}$$

where f now presents the maximum possible event rate of the inputs (which is often the clock

rate) and $P_{0\to1}$ the probability that a clock event results in a 0→1 (or power-consuming) event at the output of the gate. $C_{EFF} = P_{0\to1}C_L$ is called the *effective capacitance* and it represents the average capacitance switched every clock cycle. For our example, an activity factor of 10% ($P_{0\to1} = 0.1$) reduces the average consumption to 6W.

Low Energy-Power Design Techniques:

With the increasing complexity of digital integrated circuits, it is anticipated that the power problem will worsen in future technologies. This is one of the reasons that lower supply voltages will continue to be attractive. **ReducingV_{dd} has a quadratic effect on** P_{dyn}. For instance, reducing V_{dd} from 0.9 V to 0.45 V in our example drops the power dissipation from 6W to 1.52 W. This assumes that the same clock rate can be sustained. Experimental results [141] have shown that this assumption is not that unrealistic as long as the supply voltage is substantially higher than the threshold voltage. A larger performance penalty occurs once V_{dd} approaches $2V_{\mathrm{th}}$.

When a lower limit on the supply voltage is set by external constraints (as often happens in real-world designs) or when the performance degradation due to lowering the supply voltage is intolerable, the only means of reducing the dissipation is by lowering the effective capacitance. This can be achieved by addressing both of its components: the physical capacitance and the switching activity.

A reduction in the switching activity can only be accomplished at the logic and architectural abstraction levels and is discussed in more detail by Rabaey [141]. Lowering the physical capacitance is a worthwhile goal overall, and it also may help to improve the performance of the circuit. As most of the capacitance in a combinational logic circuit or SRAM is

due to transistor capacitances (gate and diffusion), it makes sense to keep those contributions to a minimum when designing for low power. This means that transistors should be kept to *minimal size* whenever possible or reasonable. This definitely affects the performance of the circuit, but the effect can be offset by using logic or architectural speedup techniques. The only instance where transistors should be sized up is when the load capacitance is dominated by extrinsic capacitances (such as fan-out or wiring capacitance). This is contrary to common design practices used in cell libraries, where transistors are generally made large to accommodate a range of loading and performance requirements.

These observations lead to an interesting design challenge. Assume we have to minimize the energy dissipation of a circuit with a specified lower bound on the performance. An attractive approach is to lower the supply voltage as much as possible, and to compensate the loss in performance by increasing the transistor sizes. Yet, the latter causes the capacitance to increase. It may be seen that at a low enough supply voltage, the latter factor may start to dominate and cause energy to increase with a further drop in the supply voltage.

7.6.3 Dissipation Due to Direct-Path Currents

In actual designs, the assumption of the zero rise and fall times of the input wave forms is not correct. The finite slope of the input signal causes a direct current path between V_{dd} and *GND* for a short period of time during switching, while the NMOS and the PMOS transistors are conducting simultaneously. This is illustrated in Figure 7.47.

Under the (reasonable) assumption that the resulting current spikes can be approximated as triangles and that the inverter is symmetrical in its rising and falling responses, we can

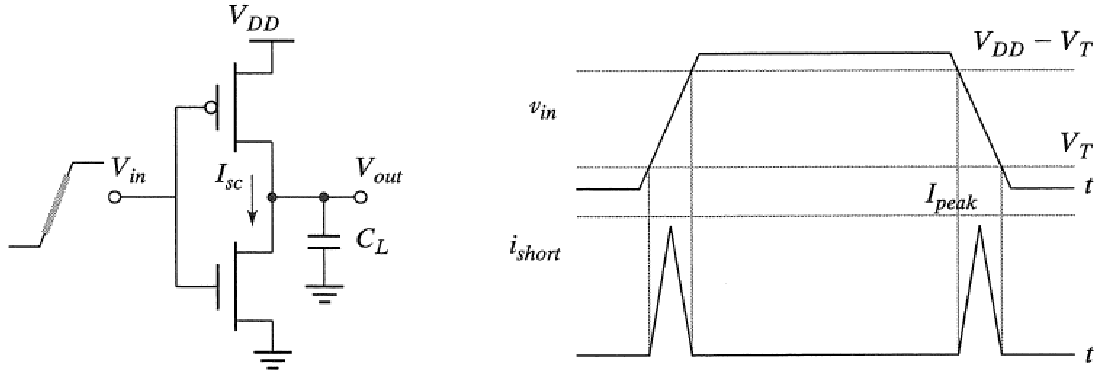


Figure 7.47: Short-circuit currents during transients [141].

compute the energy consumed per switching period as follows:

$$E_{dp} = V_{\text{dd}} \frac{I_{peak} t_{sc}}{2} + V_{\text{dd}} \frac{I_{peak} t_{sc}}{2} = t_{sc} V_{\text{dd}} I_{peak} \tag{7.35}$$

We compute the average power consumption as

$$P_{dp} = t_{sc} V_{\text{dd}} I_{peak} f = t_{sc} V_{\text{dd}} (C_{sc} \frac{V_{\text{dd}}}{t_{sc}}) f = C_{sc} V_{\text{dd}}^2 f \tag{7.36}$$

The direct-path power dissipation is proportional to the switching activity, similar to the capacitive power dissipation. t_{sc} represents the conducting time (short-circuit time) of both devices. For a linear input slope, this time is reasonably well approximated by Equation (7.37) where t_s represents the 0–100% transition time $t_{r(f)}$:

$$t_{sc} = \frac{V_{\text{dd}} - 2V_{\text{th}}}{V_{\text{dd}}} t_s \approx \frac{V_{\text{dd}} - 2V_{\text{th}}}{V_{\text{dd}}} \times \frac{t_{r(f)}}{0.8} \tag{7.37}$$

I_{peak} is determined by the saturation current of the devices and is hence directly proportional to

the sizes of the transistors. The peak current is also **a strong function of the ratio between input and output slopes**. For example, considering a static CMOS inverter with a 0→1 transition at the input, a small ratio of $\frac{^{slope}\text{input}}{^{slope}\text{output}}$ (i.e., $\frac{^{risetime}\text{input}}{^{falltime}\text{output}}$) < 1, due to a very large load capacitance) results in a short-circuit current that is close to zero. On the contrary, with the same 0→1 transition at the input, a large ratio of $\frac{^{slope}\text{input}}{^{slope}\text{output}}$ (i.e., $\frac{^{risetime}\text{input}}{^{falltime}\text{output}}$) > 3, due to a very small load capacitance) results in a short-circuit current that is close to the saturation current of the PMOS. The latter is clearly the worst case condition. This analysis leads to the conclusion that short-circuit dissipation is minimized by making the output rise/fall time larger than the input rise/fall time. On the other hand, making the output rise/fall time too large slows down the circuit and can cause short-circuit currents in the fan-out gates. This presents a perfect example of how local optimization and forgetting the global picture can lead to an inferior solution.

A more practical rule, which optimizes the power consumption in a global way, can be formulated as follows: The power dissipation due to short-circuit currents is minimized by matching the rise/fall times of the input and output signals. For the overall circuit, this means that rise/fall times of all signals should be kept constant within a range. Making the input and output rise times of a gate identical is not the optimum solution for that particular gate on its own, but keeps the overall short-circuit current within bounds.

As is apparent from Equation (7.37), the impact of **short-circuit current is reduced when we lower the supply voltage**. In the extreme case, when $V_{\text{dd}} < V_{thn} + |V_{thp}|$—where V_{thn} and V_{thp} are the threshold voltages of NMOS and PMOS transistors, respectively—short-circuit dissipation is completely eliminated because both devices are never on simultaneously. With threshold voltages scaling at a slower rate than the supply voltage, short-circuit power

dissipation has become less important in newer deep submicron technologies. At a supply voltage of 0.9 V and threshold around 0.18 V, an input/output slope ratio of 2 is needed to cause a 10% degradation in dissipation.

Finally, it is worth observing that the short-circuit power dissipation can be modeled by adding a load capacitance $C_{sc} = t_{sc} I_{peak} / V_{\text{dd}}$ in parallel with C_L, as is apparent in Equation (7.36). The value of this short-circuit capacitance is a function of V_{dd}, the transistor sizes, and the input/output slope ratio.

7.6.4 Static Consumption

The static (or steady-state) power dissipation of a circuit is expressed by the relation

$$P_{stat} = I_{stat} V_{\text{dd}} \tag{7.38}$$

where I_{stat} (also called I_{leak}) is the current that flows between the supply rails in the absence of switching activity.

Ideally, the static current of the CMOS inverter is equal to zero, as the PMOS and NMOS devices are never on simultaneously in steady-state operation. There is, unfortunately, a leakage current flowing through the reverse-biased diode junctions of the transistors, located between the source or drain and the substrate, as shown in Figure 7.48. This contribution used to be, in general, very small (a fraction of 1mW) and ignorable in older nodes, but has become somewhat considerable in newer nodes. For the device sizes under consideration, the leakage current per unit drain area typically ranges between 0.1–2.5 $\mu A / \mu m^2$ at room temperature. For

a die with 1 million gates, each with a drain area of $0.04\mu m^2$ and operated at a supply voltages of 0.9 V, the worst case power consumption due to diode leakage equals 90mW for the 16 nm node, which clearly is not negligible.

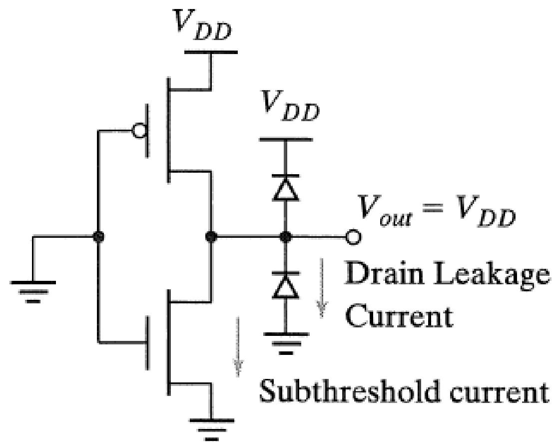


Figure 7.48: Sources of leakage currents in CMOS inverter (for V_{in}=0 V) [141].

In addition, the designer should be aware that the junction leakage currents are caused by thermally generated carries. Their values increases with increasing junction temperature, and this occurs in an exponential fashion. At $85°C$ (a commonly imposed upper bound for junction temperatures in commercial hardware), the leakage currents increases by a factor of 60 over their room-temperature values. Keeping the overall operation temperature of a circuit low is consequently a desirable goal. As the temperature is a strong function of the dissipated heat and its removal mechanisms, this can only be accomplished by limiting the power dissipation of the circuit or by using chip packages that support efficient heat removal.

The most important source of leakage current is the sub-threshold current of the transistors. A MOS transistor can experience a drain-source current, even when V_{GS} is smaller than the threshold voltage. The closer the threshold voltage is to zero volts, the larger the leakage

current at V_{GS} = 0 V and the larger the static power consumption. To offset this effect, the threshold voltage of the device has generally been kept high enough: standard processes feature V_{th} values that are never smaller than 0.180–0.200 V and, in some cases, are even substantially higher (∼0.270 V).

This approach is being challenged by the reduction in supply voltages that typically goes with deep submicron technology scaling. We concluded earlier that scaling the supply voltages while keeping the threshold voltage constant results in an important loss in performance, especially when V_{dd} approaches $2V_{\text{th}}$. We mentioned that one approach to address this performance issue is to scale the device thresholds down as well. This means that the performance penalty of lowering the supply voltage is reduced. Unfortunately, the threshold voltage is bounded below by the amount of allowable sub-threshold leakage current. The choice of the threshold voltage thus represents a trade-off between performance and static power dissipation. The continued scaling of the supply voltage predicted for the next generations of CMOS technologies, however, forces the threshold voltage ever downwards, and makes sub-threshold conduction a major source of power dissipation. Process technologies that contain devices with sharper turn-off characteristics will therefore become more attractive. An example of the latter is the SOI technology whose MOS transistors have slope factors that are close to the ideal 60 mV/decade.

Leakage Current in SRAM:

The leakage current (I_{leak}) in an SRAM cell is the major contributor to the power in the SRAM cell array. Hence, the optimization of the cell structure has to consider its impact on the cell leakage. The total leakage in a cell principally consists of the *sub-threshold leakage*,

the *gate leakage*, and the *junction band-to-band tunneling leakage* through different transistors in the cell (Figure 7.49) [116].

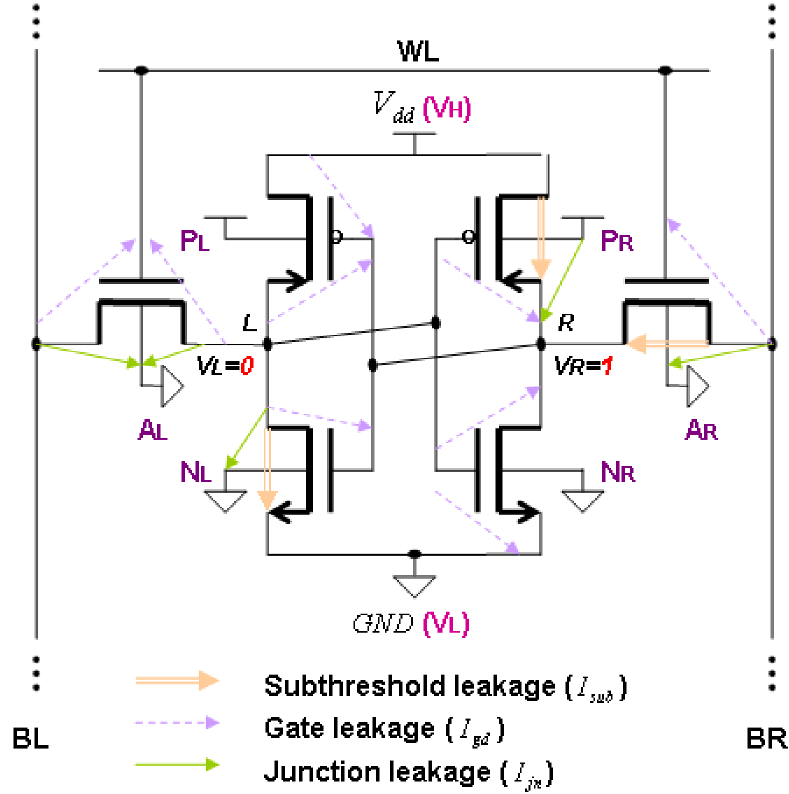


Figure 7.49: Different components of SRAM cell leakage (based on Mukhopadhyay et al. [115]).

Considering all of the different components shown in Figure 7.49, the total leakage of the cell can be computed as:

$$
\begin{aligned}
I_{sub} &= I_{subA_R} + I_{subN_L} + I_{subP_R} \\
I_{jn} &= 2I_{jnA_L} + I_{jnA_R} + I_{jnN_L} + I_{jnP_R} \\
I_{gate} &= I_{gdA_L} + I_{gsA_L} + I_{gdA_R} + I_{gdP_R} + I_{gdN_R} + I_{gsN_R} + I_{gdP_L} + I_{gsP_L} + I_{gdN_L} \\
I_{leak} &= I_{sub} + I_{jn} + I_{gate}
\end{aligned}
\tag{7.39}
$$

We can find the I_{leak} using the leakage current expressions presented in Equation (7.39) by Mukh et al. [116] to evaluate different leakage components and the total cell leakage. Alternatively, we can find the leakage current of a cell using a simplified version presented in Chapter 10 (Equation (10.6)) for estimation purposes. However, the fluctuations of the process parameters, especially fluctuations of V_{th}, result in significant variation in the leakage (particularly, the sub-threshold leakage) of the cell. In Chapter 10, we will show our simulation results for the distribution of I_{leak}.

Impact of Threshold on Performance and Static Power Dissipation

Reducing the threshold voltage by 40% multiplies the off-current of the transistors with a factor of 30! Assuming a million gate design with a 40% reduced supply voltage of 0.54V, this translates into a static power dissipation of $10^6 \times 30 \times 2.5^{-9} \times 0.54 = 40.5mW$. A further reduction of the threshold (60%) multiplies the off-current of the transistors with a factor of 160!—which results in excessive dissipation of 250mW! At that supply voltage, the threshold reductions correspond to a performance improvement of 20% and 28%, respectively.

This lower bound on the threshold is in some sense artificial. The idea that the leakage current in a static CMOS circuit has to be zero is a misconception. Certainly, the presence of leakage current degrades the noise margins, because the logic levels are no longer equal to the supply rails, but as long as the noise margins are within range, this is not a compelling issue. The leakage currents, of course, cause an increase in static power dissipation. This is offset by the drop in supply voltage, which is enabled by the reduced thresholds at no cost in performance, and results in a quadratic reduction in dynamic power. For a 16-nm CMOS process, the following circuit configurations obtain the same performance: 1.1V supply with

0.25V V_{th}; and 0.4V supply with 0.09V V_{th}. The dynamic power consumption of the latter is, however, 35 times smaller! Choosing the correct values of supply and threshold voltages once again requires a trade-off. The optimal operation point depends upon the activity of the circuit. In the presence of a sizeable static power dissipation, it is essential that non-active modules are powered down, lest static power dissipation becomes dominant. Power-down (also called standby) can be accomplished by disconnecting the unit from the supply rails or by lowering the supply voltage.

7.6.5 The Power-Delay Product, or Energy per Operation

The power-delay product (PDP) is a quality measure for a logic gate:

$$PDP = P_{avg} t_p \tag{7.40}$$

The PDP presents a measure of energy, as is apparent from the units (W s=Joule). P_{avg} is the average power. Assuming that the gate is switched at its maximum possible rate of $f_{max} = 1/(2t_p)$ and ignoring the contribution of the static- and direct-path currents to the power consumption, we find that

$$PDP = C_L V_{\text{dd}}^2 f_{max} t_p = \frac{C_L V_{\text{dd}}^2}{2} \tag{7.41}$$

Here, PDP stands for the average energy consumed per switching event (i.e., for a 0→1 or a 1→0 transition). Defining E_{avg} as the average energy per switching cycle (or per

energy-consuming event), E_{avg} thus is twice the PDP.

$$E_{avg} = 2 \times PDP = C_L V_{\text{dd}}^2 \tag{7.42}$$

7.6.6 Energy-Delay Product

The validity of the PDP as a quality metric for a process technology or gate topology is questionable. It measures the energy needed to switch the gate, which is an important property. For a given structure, however, this number can be made arbitrarily low by reducing the supply voltage. From this perspective, the optimum voltage to run the circuit would be the lowest possible value that still ensures functionality. This comes at the expense of performance, as discussed earlier. A more relevant metric should combine a measure of performance and energy. The energy-delay product (or EDP) does exactly that:

$$EDP = PDP \times t_p = (P_{avg} t_p) \times t_p = P_{avg} \times t_p^2 \tag{7.43}$$

$$EDP = PDP \times t_p = \left(\frac{C_L V_{\text{dd}}^2}{2} \right) \times t_p \tag{7.44}$$

It is worth analyzing the voltage dependence of the EDP. Higher supply voltages reduce delay, but harm the energy, and the opposite is true for low voltages. An optimum operation point should therefore exist. Assuming that NMOS and PMOS transistors have comparable threshold and saturation voltages and the devices remain in velocity saturation, we can write a

simplified version of the propagation delay expression [141]:

$$t_p \approx \frac{\theta C_L V_{\text{dd}}}{V_{\text{dd}} - V_{Te}} \tag{7.45}$$

where $V_{Te} = V_{\text{th}} + V_{DSAT}/2$ and θ is a technology parameter. Combining Equations (7.44) and (7.45) yields

$$EDP = \frac{\theta C_L^2 V_{\text{dd}}^3}{2(V_{\text{dd}} - V_{Te})} \tag{7.46}$$

The optimum supply voltage can be obtained by taking the derivative of Equation (7.46) with respect to V_{dd}, and equating the result to 0. The result is

$$V_{DDopt} = \frac{3}{2} V_{Te} \tag{7.47}$$

The remarkable outcome from this analysis is the low value of supply voltage that simultaneously optimizes performance and energy. For submicron technologies, with thresholds in the range of 0.220V, the optimum supply is situated around 0.55V.

To get a better sense for this analysis we can look at an example: From the technology parameters for our generic 16-nm CMOS process presented in this thesis, the value of V_{Te} can be derived as follows:

$V_{thn} = 0.22V$, $V_{Dsatn} = 0.32V$, $V_{Ten} = 0.38V$

$V_{thp} = -0.19V$, $V_{Dsatp} = -0.46V$, $V_{Tep} = -0.42V$

$V_{Te} \approx (V_{Ten} + |V_{Tep}|)/2 = 0.40V$

Hence, $V_{DDopt} = (3/2) \times 0.40V = 0.60V$. The simulated graphs of Figure 7.50, which plot normalized delay, energy, and energy-delay product, confirm this result. The optimum supply voltage is predicted to equal 0.58V. The charts clearly illustrate the trade-off between delay and energy.

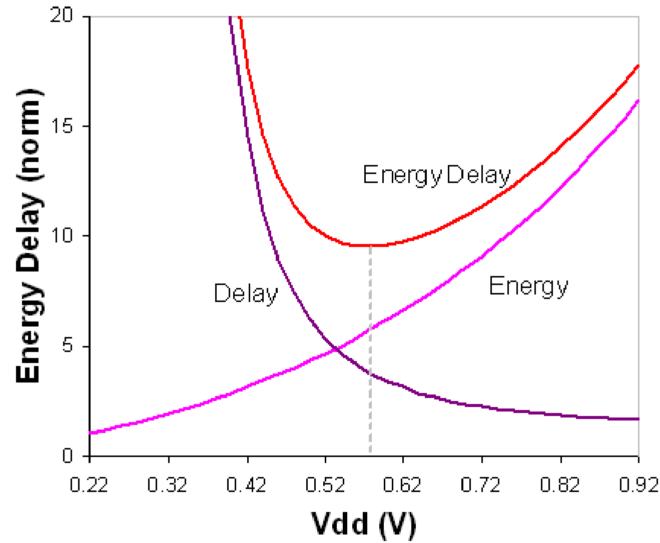


Figure 7.50: Normalized delay, energy, and energy-delay plots for CMOS inverter in 16-nm CMOS technology.

A word of caution: While the preceding example demonstrates that a supply voltage exists that minimizes the energy-delay product of a gate, this voltage does not necessarily represent the optimum voltage for a given design problem. For instance, some designs require a minimum performance, which requires a higher voltage at the expense of energy. Similarly, a lower energy design is possible by operating at a lower voltage and by obtaining the overall system performance through the use of architectural techniques such as pipelining or concurrency. For the SRAM design presented in this thesis, we choose the former mainly because our primary goal in the design was to maximize the yield with high speed and the lowest pos-

sible variability. However, in our design, we used the power reduction techniques discussed throughout this section to keep the energy consumption sufficiently low to the extent possible.

Part IV

Failure in SRAM

Chapter 8

Failure in SRAM

The sensitivity of circuit parameters to different types of variation, namely *Operational* (e.g., voltage, temperature, negative bias temperature instability (NBTI), hot carrier injection (HCI), etc, all discussed in Part III (Chapter 6 and 7), *Fabrication* (e.g., global and local process variation), and *Implementation* (e.g., physical parasitic effects, power integrity connectivity effects, layout dependent effects such as V_{th} and L_{eff}) increases as the supply voltage and feature size of semiconductor devices are reduced. This limits circuit operation in the low voltage regime, particularly for SRAM cells where submicron-sized transistors are often used [92, 119, 22]. The elevated limitation in SRAM operation increases the chances of failure in SRAM. The minimized transistors in SRAM cells are vulnerable to inter-die (D2D) as well as intra-die (also called within-die) (WID) process variation. WID process variations include random dopant fluctuation (RDF) and line edge roughness (LER), to name a few. This may result in a threshold voltage mismatch between the adjacent transistors in a memory cell giving asymmetrical characteristics [92, 115]. Moreover, it is predicted that embedded cache memo-

ries, which are expected to occupy significant portion of the total die area, will be more prone to failures with scaling [92, 144].

For successful low voltage SRAM operation, various bit-cell topologies with 5 transistors (5T-cell), 6 transistor (6T-cell), 8 transistors (8T-cell), or 10 transistors (10T-cell) have been proposed [13, 36, 30, 122, 168, 25, 35, 92, 12]. Whereas the elimination of one transistor in the 5T-cell case is proposed to lower the bitline leakage and area, the addition of extra transistors to the conventional 6T-cell is proposed to separate the *read* and the *write* mechanism. J. Kulkarni et al. [92] use a built-in feedback mechanism which incorporates process variation tolerance for improving the stability of their proposed low-power 10T-cell at the cost an increase in area. N. Azizi et al. [12] propose asymmetric dual-V_{th} for their 6T-cell design to lower the leakage power while maintaining a low access-time at the cost of stability in some cases. However, in this thesis, we stick with the conventional 6T-cell because it is the most frequently used cell in any design using on-chip memory. Our 6T-cell-based models can also be used for non-conventional SRAMs such as 5T-cell, asymmetric 6T-cell, 8T-cell, and 10T-cell, but only after some modifications, not presented in this thesis. The deviation from the fundamental conflicting design requirement of *read* versus *write* operation of a conventional 6T-cell leads to either *read* or *write failure*. Prediction of parameter fluctuations in SRAM design is a must for future nano-scaled technology nodes.

Of all the different causes of variation discussed in Part III (Chapter 6 and 7) (Figure III-A)—especially those resulting in electrical property mismatch of the different transistors in an SRAM cell—*on-die variation in the process parameters* is the most influential in SRAM failure.

Since these failures are caused by variation in the device parameters, these are known as parametric failures [113, 114]. There can also be hard failures (caused by open or short circuits) or soft failures (due to soft error). In this thesis, we will concentrate only on the parametric failures which will hereafter be referred to as "failures."

Therefore, after defining these failures in the beginning of Section 8.1, we will briefly discuss the mechanisms of each of these failures in Sections 8.1.1 through 8.1.4.

8.1 SRAM cell failure

As Figure 8.1 shows, the 6T-SRAM cell consists of two N-type access transistors and two cross-coupled CMOS inverters. Large mismatches in transistor strength due to scaling or fluctuations in the die electrical characteristics (e.g., threshold voltage, channel length, channel width, or supply voltage) can cause the cell to fail.

SRAM cell failures can be classified into four categories: *read*, *write*, *access*, and *hold* failures.

- Destructive *read*—i.e., flipping of the stored data in a cell while reading—known as *read failure*.

- Unsuccessful *write*—i.e., inability to write to a cell—defined as *write failure*.

- An increase in the access-time of the cell—i.e., resulting in a violation of the delay requirement—defined as *access failure*.

- The destruction of the cell content in standby mode with the application of a lower supply voltage—i.e., primarily to reduce leakage in standby mode—known as *hold failure*.

8.1.1 Read Failure

While reading the cell shown in Figure 8.1 (V_L = "0" and V_R = "1"), due to the voltage divider action between A_L and N_L, the voltage at node L (V_L) increases to a positive value V_{READ}. If V_{READ} is higher than the trip point of the inverter $P_R - N_R$ (V_{TRIPRD}), then the cell flips during the read [115]. This represents a read-failure event.

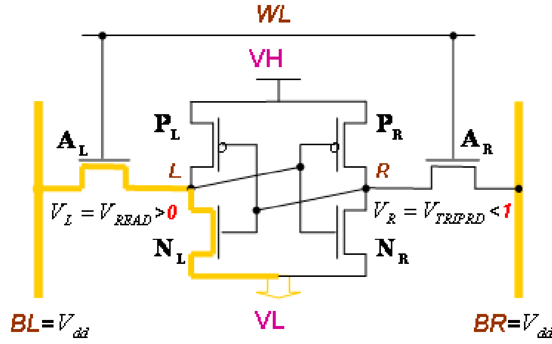


Figure 8.1: Read Failure: Flipping data during "*read*."

If the strength of the access transistor (A_L) is higher than that of the pull-down NMOS transistor (N_L), the voltage division action between the two transistors increases the voltage V_{READ}. A measure of the relative strength of the A_L and N_L is the ratio of the "ON" current (known as the beta ratio ($BR_{npd-nax}$)) of these two transistors and is given by

$$BR_{npd-nax} = \frac{\beta_{npd}}{\beta_{nax}} = \frac{\frac{\mu_{eff} C_{ox} W_{npd}}{L_{npd}}}{\frac{\mu_{eff} C_{ox} W_{nax}}{L_{nax}}} \tag{8.1}$$

where μ_{eff} is the effective mobility, Cox is the oxide capacitance (assumed to be the same as the oxide thickness if both the transistors are same), W_{npd} and W_{nax} are the widths of the pull-down and access transistor NMOS, respectively, and L_{npd} and L_{nax} are the lengths of the

pull-down and the access transistor NMOS, respectively. A decrease in $BR_{npd-nax}$ increases V_{READ}, thereby facilitating a *read failure*. Hence, while designing an SRAM cell, the size of the access transistor is usually reduced from that of the pull-down NMOS to increase $BR_{npd-nax}$. However, such a design strategy does not consider the effect of the random variation in the strengths of different transistors. For example, due to the random variation in the threshold voltage (and/or LER), a reduction in the V_{th} of the access transistor (increase in strength) and an increase in the V_{th} (reduction in strength) of the pull-down NMOS results in an increase in V_{READ} from its nominal value (i.e., value designed by optimizing the beta ratio), thereby resulting in a *read failure*. Similarly, the trip point of the inverter $P_R - N_R$ depends on the strengths of the pull-up PMOS and pull-down NMOS. Under nominal conditions, the cell is designed to have a weaker PMOS (to facilitate writing, as explained in the next section) that results in a lower value of V_{TRIP}. Although the nominal value of V_{TRIP} is not less than the nominal value of V_{READ}, parameter variation can result in an increase in the V_{th} (and/or L) of P_R and/or a reduction in the V_{th} (and/or L) of N_R. This can lower V_{TRIP} below V_{READ}, thereby resulting in *read failure*. It should be noted that the *read failure* is caused by the mismatch in the strength of the different transistors (e.g., if strength of A_L increases, while that of N_L reduces). This mismatch can only be caused by the effect of random WID variation and not by the D2D variation (which will shift the threshold voltage of all the transistors in the same direction). Hence, an increase in the random WID variation can significantly increase the *read failure*.

8.1.2 Write Failure

While writing a "0" to a cell storing "1," the V_R node gets discharged through BR to a low value (V_{WR}) determined by the voltage division between the PMOS P_R and the access transistor A_R [10]. If V_R cannot be reduced below the trip point of the inverter $P_L - N_L$ (V_{TRIPWR}) when the wordline is high (T_{WL}), then a *write failure* occurs (Figure 8.2).

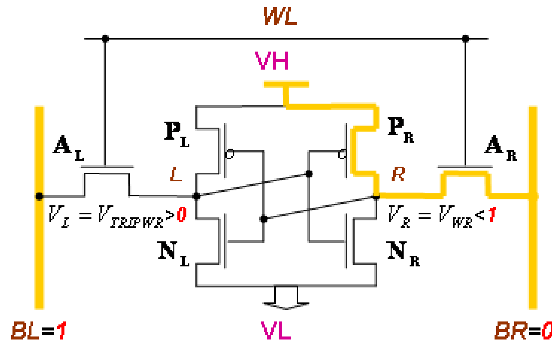


Figure 8.2: Write Failure: Memory cell does not register an input change correctly.

The discharging current (I_R) at node R is the difference in the ON currents of the access transistor A_R (I_{AR}) and the PMOS P_R (I_{PR}) (i.e., $I_R = I_{AR} - I_{PR}$). Hence, a stronger PMOS and a weaker access transistor can significantly slow down the discharging process, thereby causing a *write failure*. Thus, while designing the cell, the beta ratio between the access transistor and the PMOS ($BR_{nax-pup} = \frac{\beta_{nax}}{\beta_{pup}}$) needs to be designed (by upsizing the access and downsizing the pull-up transistors) in such a way ($BR_{nax-pup} > 1$) that under nominal conditions, the *write*-time is less than the wordline turn-on time. However, the variation in the device strengths due to random variations in process parameters can increase the *write*-time. For example, if V_{th} (and/or L) of P_R decreases and that of A_R increases, it can result in an increase in the *write*-time thereby causing a *write failure*. Hence, a proper static beta-ratio is not sufficient to reduce the

write failure. Moreover, upsizing the access transistor and/or downsizing the PMOS transistor increases the *read failure*. Thus, optimizing the size of the different transistors (considering the parameter variation) is necessary to reduce the *read* and the *write failures*. It should be noted that the *write failure* is also primarily caused by the mismatch in the strength in the transistors in a cell.

8.1.3 Access Failure

The cell *access* time (T_{ACCESS}) is defined as the time required to produce a pre-specified voltage difference ($\Delta_{MIN} \approx 0.1V_{\text{dd}}$) between two bitlines (bit-differential). If due to V_{th} (and/or L and/or V_{dd}) variation, the *access* time of the cell is longer than the maximum tolerable limit (T_{LIMIT}), an *access failure* is said to have occurred (Figure 8.3).

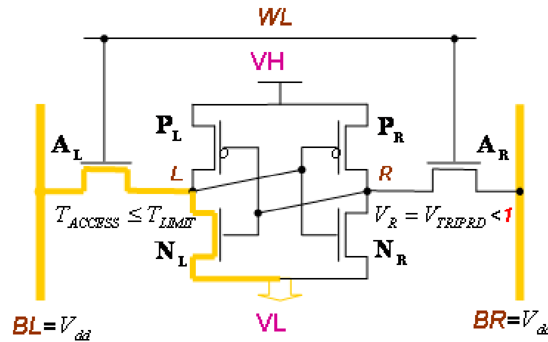


Figure 8.3: Access failure: $T_{ACCESS} > T_{LIMIT}$.

Access failure is caused by the reduction in the strength of the access and the pull-down transistors. Thus, *access failure* limits the reduction in the size of the access transistor (required to increase $BR_{npd-nax}$ to reduce V_{READ}). An increase in the V_{th} (and/or L and/or a decrease in V_{dd}) of the access transistor and the pull-down NMOS (caused by process variation)

can significantly increase the *access* time from its nominal value, thereby resulting in an *access failure*. It should be noted that as opposed to the other three types of failures, the *access failure* is caused by an increase in the V_{th} (and/or L and/or a decrease in V_{dd}) of A_L and/or of N_L. Thus, both WID and D2D variations can increase the *access failure*. Thus, *access failure* is the worst type of failure in SRAM.

8.1.4 Hold Failure

In the stand-by mode, the V_{dd} of the cell is decreased to reduce the leakage power consumption. However, if lowering the V_{dd} causes the data stored in the cell to be destroyed, then the cell is said to have failed in the *hold* mode [140] (Figure 8.4).

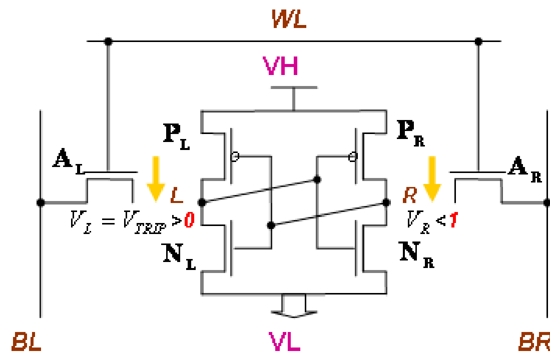


Figure 8.4: Hold failure: The destruction of the cell content in standby mode.

As the supply voltage of the cell is lowered, the voltage at the node storing “1” (node R in Figure 8.4) is also reduced. Moreover, for a low supply voltage (when P_R is not strongly “ON”), the leakage of the pull-down NMOS N_R reduces the voltage at node R, even below the supply voltage applied to the cell. If the voltage at node R is reduced below the trip-point of the inverter $P_L - N_L$, then flipping occurs and the data are lost in the *hold* mode. The supply

voltage in the *hold* mode is chosen to ensure the stability of the data under nominal conditions. However, variation in process parameters can result in device mismatch causing *hold failure*. For example, if the V_{th} (and/or L) of N_R decreases while that of P_R increases (which facilitates the reduction of the voltage at node R from the supply voltage) and/or V_{th} (and/or L) of N_L increases while that of P_L decreases (increase in the trip-point of $P_L - N_L$), the possibility of data flipping in the *hold* mode increases. Consequently, an increase in the random WID variation can significantly increase the *hold failure*.

Read, *write*, *access*, and *hold failure* probabilities, which are highly sensitive to V_{th} variation [67] and considerably sensitive to L and V_{dd} variation [115], can be as high as 5×10^{-3} for the 16-nm process.

8.2 Modeling Timing Errors

As we move to 16-nm technology and below, designing SRAMs for the worst-case parameter values will be unacceptable. Instead, SRAMs will need to be designed at the nominal-value parameters, inevitably resulting in some sections of the chip being unacceptably slower than the chip‘s frequency. In this case, the result will likely be timing faults due to variation-induced slow paths. In this section, we extend the parameter variation framework to model timing errors in SRAM critical paths due to parameter variation. We call the model VAR-TX (explained in Chapter 9). In the following, we first illustrate our general approach, describe our assumptions, and then model the errors in the critical path (composed of row-decoder, precharge, 6T-array, column-decoder, senseamp, and output-driver). We present our empiri-

cal validation of the model in Chapter 10.

8.2.1 Our General Approach and Assumptions

An SRAM typically has a multitude of paths, each one with its own time variation window, which is dependent on the input data values and output loading. In our analysis, we make two simplifying assumptions.

Assumption 1. *A path causes a timing fault if and only if it is exercised and its delay exceeds the clock period.*

Assumption 2. *A critical path is tightly designed. This means that, in the absence of process variation, there is at least one path whose delay for a certain input data value and output loading equals the clock period.*

In the following, path delay is normalized by expressing it as a fraction t_R of the pre-variation clock period t_0.

Let us first examine the probability density function (PDF) of the normalized path delays in an SRAM. Figure 8.5(a) shows an example PDF before variation effects. The right tail abuts the X = 1 abscissa and there are no timing errors.

As the SRAM critical paths suffer parameter variation, the PDF changes shape: the curve may change its average value and its spread (e.g., Figure 8.5(b)). All the paths that have become longer than 1 generate errors. Our model estimates the probability of error at a given clock period (P_E (t_R)) (also called failure probability, P_F) as the area of the shaded region in the figure. The same error probability can be obtained by generating the cumulative distribution

function (CDF), and observing that:

$$P_F = P_E(t_R) = 1 - CDF(t_R) \tag{8.2}$$

For example, Figure 8.5(c) shows the CDF of Figure 8.5(b), and the thick segment is $P_E(t_R)$ at $t_R = 1$. The CDF approach of Equation (8.2) allows for fast evaluation of the error probability at a variety of frequencies.

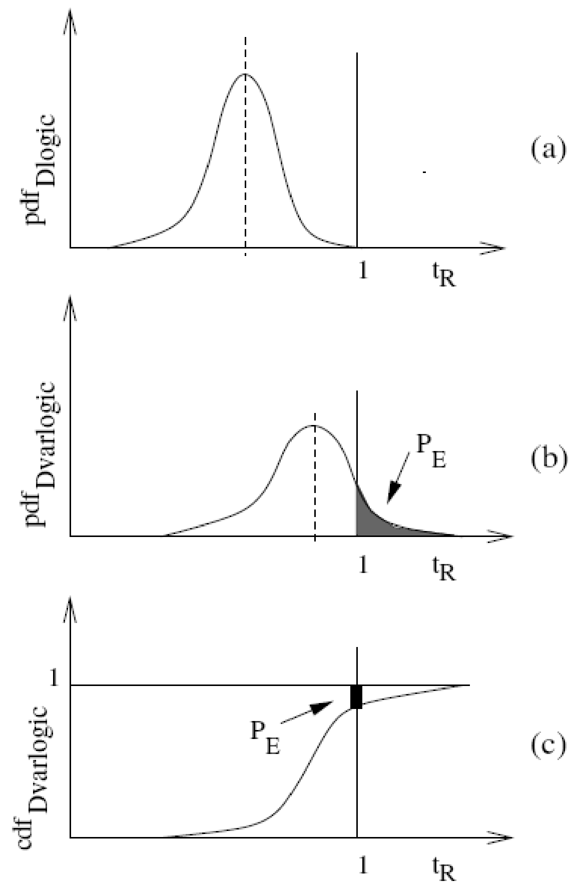


Figure 8.5: Example probability distributions.

The failure probabilities are estimated using a 6T-SRAM designed with bulk CMOS

transistors of 16-nm gate length (with $L_{\text{eff}} \approx 8nm$). The transistors are designed using the two dimensional Gaussian doping profiles [10] and simulated using the device simulator Ultrasim. In our analysis, we have used the short-channel MOSFET theory to model the delay, current, and the threshold voltage considering the device geometry and doping profile [28, 115, 33].

While modeling the failure probabilities, the random variation of V_{th}, L, and V_{dd} in the six transistors of the SRAM cell are considered six independent Gaussian random variables (mean = 0) [22]. The assumption of the independent random variable is justified as we have considered the effect of RDF for V_{th}, LER for L, and residual IR-Drop and/or Ldi/dt (resulting in very small change in V_{dd}).

For example, the placement and the number of dopants in the channel of one transistor depend only on the geometry of that transistor and are independent of the placements and the number of dopants in the channel of a neighboring transistor [21]. Thus, the V_{th} fluctuation due to the RDF of one transistor does not depend on the V_{th} fluctuation of any neighboring transistor. Hence, the ΔV_{th} of the cell transistors can be assumed to be independent random variables [21]. Similarly (although to a lesser degree), the shape of the edges of the channel of one transistor along its length (and width) depends only the geometry of that transistor and is independent of the shape of the edges of the channel of a neighboring transistor. A similar justification holds for V_{dd} as well - as the actual V_{dd} applied to one transistor could be slightly different than that of a neighboring transistor due to different nearby coupling capacitances.

The standard deviation of the ΔV_{th} fluctuation (δV_{th}) due to RDF depends on the manufacturing process, doping profile, and the transistor sizing [115]. In the proposed method, δV_{th} for a minimum-sized transistor (δV_{th0}) is an input parameter and the dependence of δV_{th}

on the transistor size is given by [115].

$$\sigma V_{\text{th}} = \sigma V_{\text{th0}} \sqrt{\left(\frac{L_{min}}{L}\right)\left(\frac{W_{min}}{W}\right)} \tag{8.3}$$

In addition to the random variation, we have also investigated and incorporated the effect of the systematic correlation of ΔV_{th}, ΔL, and ΔV_{dd} on the failure probabilities in our timing error analysis, as explained in the next sub-section.

8.2.2 Timing Errors in SRAM Memory

To model variation-induced timing errors in SRAM memory, we build upon and extend the work of Mukhopadhyay et al. [115]. Whereas these authors describe the four types of SRAM failure considering only *random* V_{th} variation, we consider the combination of *random* and *systematic* V_{th}, L, and V_{dd} variation when describing the four types of failures in an SRAM cell (Figure 8.1 - 8.4). However, we confirm that *random* V_{th} variation is the major source of WID variation simply because of the small geometry of the SRAM cell. The principal source of the device mismatch is the intrinsic fluctuation of the V_{th} of different transistors due to RDF [22]. As the transistors in a cell are in very close spatial proximity, the effect of mismatch in the channel length or width or, particularly, in the supply voltage is small.

Since *access failures* dominate the four types of failure [115, 169] (as explained in Section 8.1.3), we focus on modeling *access* errors only.

To find the *access* timing errors (and therefore the timing errors in SRAM), we first find the total path delay D_p (*access* time of SRAM) distribution, as explained in detail in Chap-

ter 9.

Once we have the total path delay D_p (access-time) distribution, we numerically integrate it to obtain its CDF_{Dp} (Figure 8.5(c)). Then, the probability failure of SRAM is the estimated error rate P_E of the path D_p cycling with a relative clock period t_R.

$$P_F = P_E(t_R) = 1 - CDF_{Dp}(t_R) \tag{8.4}$$

We will use this P_F in Chapter 9 to compute the yield.

Part V

Proposed Model: VAR-TX

Chapter 9

Our Proposed Model

In this chapter, we present a new method for path-based statistical timing analysis that takes both the process variations and the architectural aspect of SRAMs into consideration for delay optimization and, therefore, yield enhancement. We first propose a method for modeling D2D and WID device parameter variations. Based on this model, we then present an efficient method for computing the total path delay probability distribution using a combination of device parameter enumeration for D2D and an analytical approach for WID variation. We employ a simple and effective model of spatial correlation of WID device parameter variation introduced by Agarwal [4] to extend our analysis. We test the proposed method on our designed industrial high-performance 6T-SRAM and present comparisons with traditional path analyses which do not distinguish between different architectures/organizations of SRAM. We show that the proposed statistical analysis can significantly improve the accuracy of performance modeling. We validate our computed total path delay probability distribution with that of VARIUS [169] and Mukh [115] and demonstrate the accuracy of the proposed approach by comparing our results

with Monte-Carlo simulation. The characteristics of the device parameters, as well as assumptions about their associated spatial correlation, are presented in Sections 9.1 and 9.2. A rather shorter version of our modeling methodology is discussed in our ISQED–2012 published paper [147] presented in Appendix A.

In recent technologies, the variability of circuit delay due to process variation has become a significant concern. As process geometries continue to shrink, the ability to control critical device parameters is becoming increasingly difficult, and as a result, there are significant variations in device length, doping concentrations, and oxide thicknesses. These process variations pose a significant problem for timing, as well as yield prediction, and require that static timing analysis models the circuit delay not as a deterministic value, but as a random variable.

Using static timing analysis has become the primary method for performance verification of high performance designs. Static timing analysis has the advantage that it does not require input vectors and has a run time that is linear with the size of the circuit.

In a path based approach, deterministic timing analysis is first performed and the top n critical paths are enumerated, where n is a sufficiently large number to include all paths that have a significant probability of being critical. For instance, if the delay variability is expected to be 10% of nominal, all paths that have a deterministic delay within 10% of the worst-case circuit delay must be included. The delay of each path is then statistically analyzed resulting in a probability distribution. The 3-sigma delay (or any other desired confidence point) is then computed for each path and is compared against the required circuit performance. This approach avoids the issue of path reconvergence, thereby simplifying the problem and allowing for the use of more accurate models. Path-based statistical timing analysis provides statistical

information on a path-by-path basis. It accounts for WID process variation and hence eliminates the pessimism in deterministic timing analysis based on case files. It also accurately indicates which paths are critical under process variability, allowing for better optimization of the circuit.

Due to its close relevancy to our modeling strategy, we discussed, among others, the notion of *systematic* and *random* variation, as well as *D2D* and *WID* classifications in Section 6.1) to set the tone for the discussion and/or the formulas presented in this chapter.

As explained in Chapter 8, among the four types of SRAM failures—(*Read failure*, *Write failure*, *Access failure*, and *Hold failure*)—*Access failure* is by far the most influential culprit for chip failure [115]; therefore, we only consider Access Failure in our analysis of SRAM delay and delay variations. However, the minimum clock period that we choose for each technology node (16-nm, 45-nm, and 180-nm) is determined in such a way to avoid all four types of SRAM cell failures up to the failure probability specified by the International Technology Roadmap for Semiconductors (ITRS) [1]. The clock period is the summation of precharge time, read/write time, and senseamp and output-driver component delays managed by the internal-clock circuitry—which sets the slew rate and pulse width for the timing of SRAM components. These timings are empirically tested, adjusted, and verified to ensure sufficient SNM and stability, and therefore, robust memory design.

Model stage delays depend on input slopes and output loading. This means different SRAM architectures/organizations exhibit different component delays. For example, assuming all other parameters are equal, the precharge component delay in an architecture that uses a larger row decoder, and therefore longer bitlines, exceeds the precharge delay in an architecture that uses a smaller row decoder and shorter bitlines. This is because the input rise time and

output loading of the precharge in the former are larger than in the latter. Of course, this is only a valid statement if all other parameters, such as word-size, are the same for the two architectures.

Similarly, different organizations exhibit differing SRAM component power and area consumption. For example, a given large-sized SRAM that has only one bank consumes more power but less area as compared to the same large-sized SRAM that has been divided into several banks. (This is because all components of a single-bank SRAM, having smaller area, are active while all components in the non-active banks of the multiple-bank SRAM, spanning larger area, are put to sleep).

We measured and recorded the delay and delay variation for each stage of each SRAM component. The stage delays and variations were then combined to calculate total component delays and total component delay variations. Similarly, the power (and variation in power) for each sub-circuit of each SRAM component is measured, recorded, and then combined. For area measurement, we use layout extrapolation methodology (similar to tiling style). This chapter first details how we measure and combine the delays, and how these delays are used to obtain access-times and variations of access time. Then, we explain the computation/estimation of leakage current, static and dynamic power, and their associated variations, as well as the estimation of area. Lastly, we explain our new VAR-TX model.

9.1 Derivation of access-time and its variation

The proposed model and analysis method was applied to the variation in all three major device parameters (V_{th}, L, and V_{dd}) and for all different feasible 6T-SRAM architectures. We obtained D2D and WID device parameter variation from predictions of the (ITRS) [1] and the experimental data of other published work [169]. For spatial correlation component allocations, we used our own empirically collected data.

To compute the WID path delay component of process variability, we first compute the sensitivity of gate delay, output slope, and input load with respect to the input slope, output load and device parameters for all feasible architectures. Using these sensitivities, we then express the path delay variation as an analytical expression of the device parameter variation, allowing for very efficient analysis of WID variability, including an accurate model for spatial correlation. Since the D2D component of the path delay variability is dependent on a single random variable, we can compute it efficiently through enumeration of its probability distribution. We then compute the joint path delay distribution through the convolution of WID and D2D delay distribution components to obtain the distribution of the total delay variability.
Here is our derivation process for delay distribution in abstract:

1. Compute the sensitivities and store them in tables.
2. Compute the D2D component of the path delay.
3. Express the WID component of the path delay variation as an analytical expression of the device parameter variation.

4. Combine the two components (namely, D2D and WID) of the path delay variations to obtain the joint path delay distribution.

5. Optimize the delay through the examination of all possible architectures to achieve maximum yield.

In order to extract the total delay variation due to the device parameter variation, we use a first-order approximation which is widely used in statistical timing analysis [4, 192, 128], shown in Equation (9.1):

$$P_{total,i} = P_0 + \Delta P_{D2D} + \Delta P_{WID,i} = P_{D2D} + \Delta P_{WID,i} \quad (9.1)$$

where P represents any of the three parameters in the system, such as V_{th}. We model each device parameter $P_{total,i}$ of device i as the algebraic sum of a D2D device parameter P_{D2D} and a WID device parameter variation $\Delta P_{WID,i}$. The D2D device parameter is defined as $P_{D2D} = P_0 + \Delta P_{D2D}$, where P_0 is the nominal value of P, and ΔP_{D2D} is the change in the delay of a device due to D2D variation

We can apply this generic equation to the specific device parameters that we consider in our model—V_{th}, L, and V_{dd}:

$$V_{\text{th total, i}} = V_{\text{th D2D}} + \Delta V_{\text{th WID, i}} \quad (9.2)$$

$$L_{\text{total, i}} = L_{\text{D2D}} + \Delta L_{\text{WID, i}} \quad (9.3)$$

$$V_{\text{dd total, i}} = V_{\text{dd D2D}} + \Delta V_{\text{dd WID, i}} \quad (9.4)$$

For each device parameter P, all devices on a die share one variable P_{D2D} for the D2D component of their $P_{total,i}$, which represents the *due-to-P* mean of the gate of a particular die (e.g., $V_{\text{th D2D}}$, L_{D2D}, and $V_{\text{dd D2D}}$ represent the mean of all devices on a die with respect to V_{th}, L, and V_{dd}, respectively). The WID component of each device has a separate independent random variable $\Delta P_{WID,i}$, where all random variables $\Delta P_{WID,i}$ have identical probability distributions (e.g., each device on a die has a separate independent random variable $\Delta V_{\text{th WID, i}}$, $\Delta L_{\text{WID, i}}$, and $\Delta V_{\text{dd WID, i}}$ that are different than those of the neighboring devices). The D2D variation P_{D2D} has a mean which is equal to the nominal value of P of device. The WID variation $\Delta P_{WID,i}$ has systematic and random components of which the latter has a mean of zero. The total variation P_{total}, therefore, has a mean equal to sum of the mean of P_{D2D} and the mean of $\Delta P_{WID,i}$. We assume that all three random variables $P_{total,i}$, P_{D2D}, and $\Delta P_{WID,i}$ have a normal distribution, which is a common assumption since device threshold voltage, length, and supply voltage are physical quantities.

We obtain the distribution of the path delay D_p, resulting from the variation of all device parameters and delay of the individual gates in the path of a certain architecture, through Equation (9.5). The path delay D_p is a random variable, D_i is the delay of gate i as a function of its device parameters, and the sum is taken over all gates of a path of certain architecture.

$$D_p = \sum_{i}^{n} D_i(P_{D2D} + \Delta P_{WID,i}) \tag{9.5}$$

The computation of the D_p distribution is difficult since D_i is a non-linear function that cannot be accurately expressed in closed form. Therefore, we resort to two feasible methods. One

method for computing the distribution of D_p is through Monte-Carlo simulation that we will perform in section 10.1. Another method is to use the following simplifying assumption:

$$D_i(P_{D2D} + \Delta P_{WID,i}) = D_i(P_{D2D}) + \Delta D_i(\Delta P_{WID,i}) \tag{9.6}$$

which shows that the gate delay in a certain architecture is approximated by the sum of the D2D delay and WID variation of the gate in that architecture. The assumption of Equation (9.6) allows us to compute $D_i(P_{D2D})$ and $\Delta D_i(\Delta P_{WID,i})$ independently and then combine them to obtain the total path delay distribution D_p, as follows:

$$D_p = \sum_i^n D_i(P_{D2D}) + \sum_i^n \Delta D_i(\Delta P_{WID,i}) \tag{9.7}$$

We discuss the computation of the two components of D_p in the following two sub-sections.

9.1.1 D2D variability analysis

To compute the delay due to D2D variation we need to compute $D_{p,D2D}$ as a function of D2D device parameters as in Equation (9.8).

$$\begin{aligned} D_{p,D2D} &= \sum_i^n D_i(P_{D2D}) \\ &= f\left[\sum_i^n D_i(V_{\text{th D2D}}), \sum_i^n D_i(L_{\text{D2D}}), \sum_i^n D_i(V_{\text{dd D2D}})\right] \end{aligned} \tag{9.8}$$

For each parameter (V_{th}, L, V_{dd}), the corresponding gate delay in $D_{p,D2D}$ shares a single random variable, therefore, the D2D variation of D_p due to each parameter can be computed separately

through enumeration of the distribution of V_{th}, L, and V_{dd} ($V_{\text{th D2D}}$, L_{D2D}, $V_{\text{dd D2D}}$). We enumerate the different possibilities from the worst case to the best case process corners for each of the three parameters, and compute the resulting path delay $D_{p,D2D}$ for each of the three cases individually. The probability distribution of $D_{P,D2D}$ for each individual parameter is then computed by considering the probability distribution ($V_{\text{th D2D}}$, L_{D2D}, or $V_{\text{dd D2D}}$) of the selected device parameter (V_{th}, L, or V_{dd}) and their associated resulting path delay for each enumeration. We then combine the mean and the variance of all three distributions to obtain the mean and variance of $D_{p,D2D}$. In our experiments, discretization of $V_{\text{th D2D}}$ into 30 device thresholds, L_{D2D} into 20 device lengths, and $V_{\text{dd D2D}}$ into 3 device supply voltages was sufficient to obtain a high level of accuracy. This requires simulating each path 30 times for V_{th}, 20 times for L, and 3 times for V_{dd}, for each of the feasible architectures, which is a relatively low cost for computing $D_{p,D2D}$.

9.1.2 WID variability analysis

The path delay variation due to WID device parameter variation (the second term in Equation (9.6) is a function of multiple independent random variables. Therefore, the number of simulations required for computing $D_{p,WID}$ through enumeration is Θ^n, where Θ is the number discretizations of $\Delta P_{WID,i}$ and n is the number of gates in the path. Even for paths consisting of a few gates, this approach is computationally infeasible. Therefore, we make a second simplifying assumption, namely that $\Delta D_i(\Delta P_{WID,i})$ can be approximated linearly as

$$\Delta D_i(\Delta P_{WID,i}) = \frac{\partial D_i}{\partial P_{WID,i}} \times \Delta P_{WID,i} = coef_i \times \Delta P_{WID,i} \tag{9.9}$$

for small values of $P_{WID,i}$, where the sensitivity of the delay with respect to device parameter $\partial D_i/\partial P_{WID,i}$ is computed at the nominal device parameter value. The simplification of Equation (9.9) allows us to compute the change of path delay $D_{p,WID}$ due to WID device parameter variation analytically and efficiently, using pre-computed delay sensitivities (*coefi*). When computing $D_{p,WID}$, the dependence of the delay of gate i on gate input load of its fan-out gate i+1 must be considered, which is a function of the device parameter $\Delta P_{WID,i+1}$. Similarly, the delay of gate i is dependent on its input slope, which is a function of all device parameters $\Delta P_{WID,j}$, where gate $j < i$ precedes gate i in the path. We therefore extend the linear assumption of Equation (9.9) to the change of a gate delay and output slope due to input slope and output load and formulate the computation of $D_{p,WID}$ for each parameter (V_{th}, L, and V_{dd}) in the same way, which is shown below for the threshold voltage case. The change in path delay due to V_{th} ($D_{p,WID,Vth}$) is the sum of the individual gate delay changes ΔD_i due-to-V_{th}, where each of the gate delay changes and their corresponding output slope changes are a function of the change in output slope of the *preceding* gate ΔS_{i-1}, the change in input load of the *succeeding* gate ΔCl_{i+1}, and the WID device threshold $\Delta V_{\text{th WID, i}}$:

$$\Delta D_i = f(\Delta S_{i-1}, \Delta Cl_{i+1}, \Delta V_{\text{th WID, i}}) \tag{9.10}$$

$$\Delta S_i = f(\Delta S_{i-1}, \Delta Cl_{i+1}, \Delta V_{\text{th WID, i}}) \tag{9.11}$$

The change in delay, slope, and input capacitance of a single gate is approximated as a sum of products of the sensitivities and the change in the threshold values:

$$\Delta D_i = \frac{\partial D_i}{\partial S_{i-1}} \times \Delta S_{i-1} + \frac{\partial D_i}{\partial Cl_{i+1}} \times \Delta Cl_{i+1} + \frac{\partial D_i}{\partial V_{\text{th i}}} \times \Delta V_{\text{th WID, i}} \tag{9.12}$$

$$\Delta S_i = \frac{\partial S_i}{\partial S_{i-1}} \times \Delta S_{i-1} + \frac{\partial S_i}{\partial Cl_{i+1}} \times \Delta Cl_{i+1} + \frac{\partial S_i}{\partial V_{\text{th i}}} \times \Delta V_{\text{th WID, i}} \tag{9.13}$$

$$\Delta Cl_i = \frac{\partial Cl_i}{\partial V_{\text{th i}}} \times \Delta V_{\text{th WID, i}} \tag{9.14}$$

The seven basic sensitivities of delay (ΔD_i) and slope (ΔS_i) with respect to input slope, output load and device threshold and the sensitivity of gate input load (ΔCl_i) with respect to device threshold are pre-computed for each gate over a range of output load and input slope conditions. In this thesis, we computed the sensitivities for all cases of V_{th}, L, and V_{dd} during circuit simulation by the use of the curve fitting method illustrated for V_{th} in Figure 9.1.

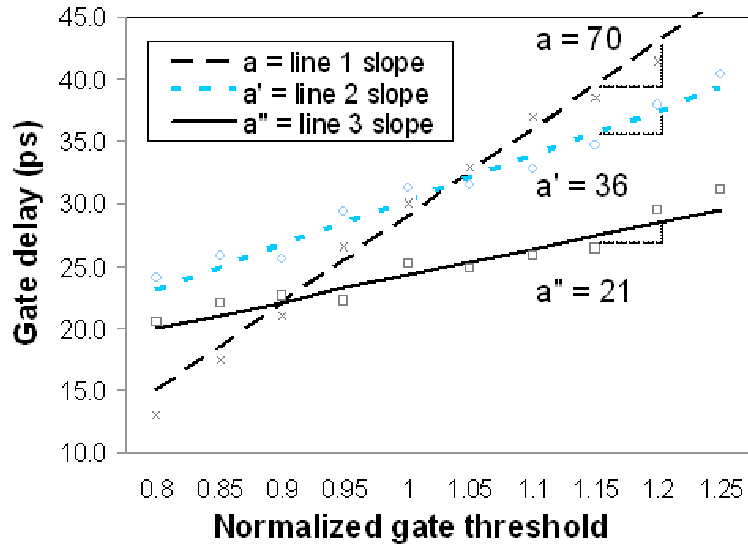


Figure 9.1: Curve fitting for Hspice simulation for an SRAM.

The delay for a gate (located on the critical path) with respect to either of the device parameters (e.g., $d_{g,WID,Vth}$) is the summation of the gate nominal delay (d_{g0}) and the additional delay caused by parameter fluctuation of the device (e.g., $\Delta d_{g0,Vth}$).

$$d_{g,WID,Vth} = d_{g0} + \Delta d_{g0,Vth} \tag{9.15}$$

For small scale fluctuations, such first-order linear approximation is accurate enough. Figure 9.1 shows the delay and gate threshold fitting curve for the computation of the sensitivity $\partial D_i/\partial V_{\text{th WID, i}}$ (represented by three different slopes a, a', a'' for three different architectures in 16-nm 64KB SRAM), and the linear fits match the Hspice simulation for the range under consideration. A similar agreement holds for L and V_{dd} cases in different architectures, as well. These basic sensitivities along with their associated WID variations ΔP_{WID} (discussed below in this section) are stored in tables and are accessed during the computation of $D_{p,WID}$ for a particular path using linear interpolation of the stored values in the table.

It is interesting to observe in Figure 9.1 how the delay and gate threshold fitting curves for the same gate can be different under different circumstances. While the upper curve (line 1) shows a larger slope (indicating larger variation) for a gate used in an 1:64:1024 (columns:word-size:rows) architecture, the bottom curve (line 3) exhibits a relatively smaller slope (indicating smaller variation) for the same gate used in a 64:64:16 architecture. The middle curve (line 2) of architecture 4:64:256 shows a slope between those of the other two. Such differences in the slope is mainly due to the different cumulative loading effect of the preceding gates on the input slope of the gate and the different loading effect of the succeeding gate on the output

capacitance of the gate in different architectures.

The access-time is the summation of the associated critical path nominal delay (D_0) and the additional delay caused by parameter fluctuations of each device on the path, assuming n total devices. Since the numerical value of the total parameter variation of each of the n gates in different architectures is different, the access-time and variation of access-time for different architectures is different, as well. In sections 9.2 and 9.3, we show how choosing the optimal architecture can reduce the access-time and/or variation of access-time.

We then combine Equations (9.12)–(9.14) to obtain an expression of ΔD_i as a function of the basic sensitivities and WID device threshold variations. The delay change coefficients of this function are efficiently computed for all gates in the path using a single traversal of the path for each architecture using the basic seven sensitivities. We then collect all the gate delay coefficients with respect to each WID device threshold and express the total change in path delay $D_{p,WID,Vth}$ (due-to-V_{th}) as follows:

$$D_{p,WID,Vth} = \sum_{i}^{n} x_i \times \Delta V_{\mathrm{th\ WID,\ i}} \tag{9.16}$$

where x_i is the coefficient of total path delay change due to WID device threshold $\Delta V_{\mathrm{th\ i}}$ at gate i. Equation (9.17) shows the total WID path delay change *due-to-all-device-parameters* for one of the m number of architectures.

$$D_{p,WID} = \sum_{i}^{n} \left(x_i \times \Delta V_{\mathrm{th\ WID,\ i}} + y_i \times \Delta L_{\mathrm{WID,\ i}} + z_i \times \Delta V_{\mathrm{dd\ WID,\ i}} \right) \tag{9.17}$$

For WID variations ΔP_{WID}, there are both correlated (systematic) and random components. To capture this effect, we use the method introduced by Agarwal [4]. The SRAM area is divided into a multi-level quad-tree partitioning as shown in Figure 9.2.

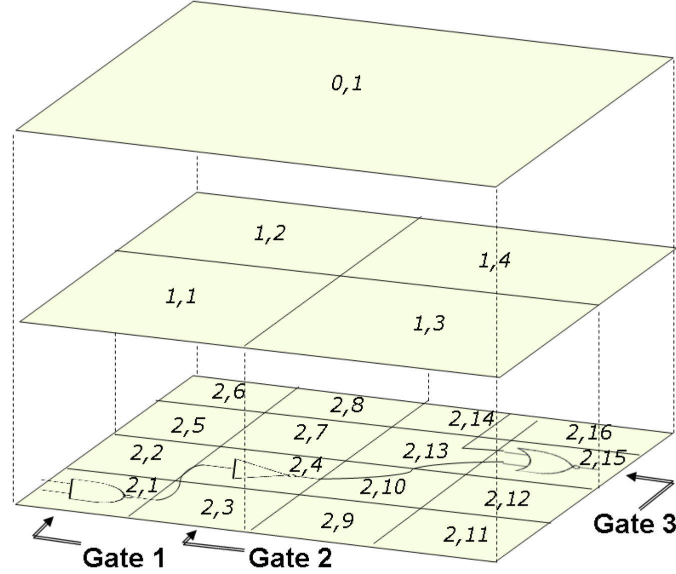


Figure 9.2: Spatial correlation modeling for WID variations (Based on Fig.1 of Agarwal [4]).

For each level , the die area is partitioned into $2^l - by - 2^l$ squares, where the first (or top) level 0 has a single region for the entire die and the last (or bottom) level k has $4k$ regions. We then associate an independent random variable $\Delta P_{l,r}$ with each region (l,r) to represent a component of the total WID device parameter variation (e.g., $\Delta P_{0,1}$, $\Delta P_{1,1}$, $\Delta P_{2,1}$). The variation of a gate i is then composed of a sum of WID device parameter components $\Delta P_{l,r}$, where level l ranges from 0 to k and the region r at a particular level is the region that intersects with the position of gate i on the die. For example, for the gate in region 2,1 in Figure 9.2, the components of WID device length variation would be $\Delta L_{0,1}$, $\Delta L_{1,1}$, and $\Delta L_{2,1}$.

Applying the generic Equation (9.18) to the specific device parameters that we con-

sider in our model—V_{th}, L, and V_{dd}—gives Equations (9.19)–(9.21).

$$\Delta P_{WID,i} = \sum_{0<l<k,\ r\ intersects\ i} \Delta P_{l,r} \tag{9.18}$$

$$\Delta V_{\text{th WID, i}} = \sum_{0<l<k,\ r\ intersects\ i} \Delta V_{\text{th l,r}} \tag{9.19}$$

$$\Delta L_{\text{WID, i}} = \sum_{0<l<k,\ r\ intersects\ i} \Delta L_{\text{l,r}} \tag{9.20}$$

$$\Delta V_{\text{dd WID, i}} = \sum_{0<l<k,\ r\ intersects\ i} \Delta V_{\text{dd l,r}} \tag{9.21}$$

Gates that lie within close proximity of each other will have many common WID device parameter components resulting in a strong WID parameter correlation. Gates that lie far apart on a die share few common components and therefore have a weaker correlation. Figure 9.2 shows an example of a die with 3 levels of partitioning resulting in 16 regions at the bottom level. Since the number of regions at the bottom level grows as 4^k it is possible to obtain a fine partitioning of the die with only a moderate number of levels. We apply 6 levels of quadrants (the same number of levels of quadrants used by Agarwal [4]) with the top quadrant the entire SRAM and the bottom quadrant the individual devices for gate parameter modeling. The 6 levels of quadrants give us a fine partitioning quite sufficient for our first order approximation.

Note also that, if the same total independent variation is assigned for the WID and D2D sigma of a parameter, the parameter (i.e., length $\Delta L_{0,1}$) associated with the region at the top level of the hierarchy will be equivalent to the D2D device parameter (i.e., length L_{D2D})

since it is shared by all gates on the die.

We can control how quickly the spatial correlation diminishes as the separation between two gates increases by correctly allocating the total WID device parameter variation among the different levels. If the total WID variance is largely allocated to the bottom levels and the regions at the top levels have only a small variance, there is less sharing of device parameter variation between gates that are far apart and the spatial correlation will diminish quickly. This yields results that are similar to an uncorrelated WID analysis. On the other hand, if the total WID variance is predominantly allocated to the regions at the top levels of the hierarchy, then even gates that are widely spaced apart will still have significant correlation. This will yield results that are close to the traditional approach, where all gates are perfectly correlated and the WID device parameter variation is zero. The quad-tree model that we have decided to use is therefore flexible and can be easily modified to measured device parameter data. Also, it is straightforward to extend the model to include topological and structural correlations, such as gate orientation.

We illustrate the spatial correlation model for the length parameter of the three gates shown in Figure 9.2 in regions (2,1), (2,4) and (2,15). For each quadrant, we generate a random variable according to a normal distribution. The WID device length variation of these gates is the sum of the device length variation components associated with the regions that the gate is

located in, leading to the following equations:

$$\Delta L_{\text{WID,Gate 1}} = \Delta L_{2,1} + \Delta L_{1,1} + \Delta L_{0,1} \tag{9.22}$$

$$\Delta L_{\text{WID,Gate 2}} = \Delta L_{2,4} + \Delta L_{1,1} + \Delta L_{0,1} \tag{9.23}$$

$$\Delta L_{\text{WID,Gate 3}} = \Delta L_{2,15} + \Delta L_{1,4} + \Delta L_{0,1} \tag{9.24}$$

We can observe from the WID device length equations that gates 1 and 2 are strongly correlated, as they share the common variables $\Delta L_{1,1}$ and $\Delta L_{0,1}$. On the other hand, gates 1 and 3 are more weakly correlated as they share only the common variable $\Delta L_{0,1}$.

We apply the same procedure for WID of V_{dd} ($\Delta V_{\text{dd WID}}$) and the systematic-WID of V_{th} ($\Delta V_{\text{th WID,sys}}$). Our $\Delta V_{\text{th WID,sys}}$ has an inverse relation to the square root of $L \times W$ [115, 193], (Equation (8.3)). We model the random-WID of V_{th} ($\Delta V_{\text{th WID,rand}}$) as a random variable which obeys a normal distribution [115] due to the random dopant effect (RDF) and line edge roughness (LER) [193]. The summation of $\Delta V_{\text{th WID,sys}}$ and $\Delta V_{\text{th WID,rand}}$ gives $\Delta V_{\text{th WID}}$. To model L and V_{dd} with highly correlated variation and V_{th} with mostly random variation (weakly correlated), we allocate most of ΔL_{WID} and $\Delta V_{\text{dd WID}}$ to the higher levels and most of $\Delta V_{\text{th WID,sys}}$ to the lower levels in the hierarchy of Figure 9.2.

The change in delay due to WID device length variation for these gates can be expressed as the product of their WID device length components with their respective coefficients of the total path delay change. Using Equation (9.18) (or specifically Equation (9.20)), we get

the following equations:

$$\Delta D_{Gate\,1} = K_1(\Delta L_{2,1} + \Delta L_{1,1} + \Delta L_{0,1}) \tag{9.25}$$

$$\Delta D_{Gate\,2} = K_2(\Delta L_{2,4} + \Delta L_{1,1} + \Delta L_{0,1}) \tag{9.26}$$

$$\Delta D_{Gate\,3} = K_3(\Delta L_{2,15} + \Delta L_{1,4} + \Delta L_{0,1}) \tag{9.27}$$

Summing up the ΔD_is in Equations (9.25) through (9.27), we get the change in the path delay $D_{p,WID,L}$ due to spatially correlated WID device length variation as follows:

$$\begin{gathered} D_{p,WID,L} = K_1(\Delta L_{2,1}) + K_2(\Delta L_{2,4}) + K_3(\Delta L_{2,15}) + \\ (K1 + K_2)(\Delta L_{1,1}) + K_3(\Delta L_{1,4}) + (k_1 + K_2 + k_3)\Delta L_{0,1} \end{gathered} \tag{9.28}$$

Following the same procedure illustrated for device length, we can also compute the path delay due to spatially correlated WID device V_{th} variation and the path delay due to spatially correlated WID device V_{dd} variation to obtain $D_{p,WID,Vth}$ and $D_{p,WID,Vdd}$, respectively.

Given the mean ($\mu_{\mathrm{Vth\ i}}$, $\mu_{\mathrm{L\ i}}$, and $\mu_{\mathrm{Vdd\ i}}$) and the standard deviation ($\sigma_{\mathrm{Vth\ i}}$, $\sigma_{\mathrm{L\ i}}$, and $\sigma_{\mathrm{Vdd\ i}}$) for the WID device threshold $\Delta V_{\mathrm{th\ i}}$, WID device length ΔL_{i}, and WID device supply voltage $\Delta V_{\mathrm{dd\ i}}$, with normal distribution and the coefficients x_i, y_i, and z_i, we can compute the mean and standard deviation of the probability distribution for $D_{p,WID}$ directly using the

following standard equations:

$$\mu_{D_{p,WID}} = \sum_{i}^{n} (x_i \times \mu_{\text{Vth i}} + y_i \times \mu_{\text{L i}} + z_i \times \mu_{\text{Vdd i}}) \tag{9.29}$$

$$\sigma_{D_{p,WID}} = \sqrt{\sum_{i=1}^{n} (x_i^2 \times \sigma^2_{\text{Vth i}} + y_i^2 \times \sigma^2_{\text{L i}} + z_i^2 \times \sigma^2_{\text{Vdd i}})} \tag{9.30}$$

Given pre-characterized sensitivities, the final computation of the distribution of $D_{p,WID}$ is performed very efficiently and requires only a single traversal of the path for each of the architectures. We show the major impact of the architecture-dependent WID variations on the access-time by comparing its distribution ($D_{p,WID}$) to the total D_p—both computed through the proposed analytical approach, and both are compared with Monte Carlo simulation—in Chapter 10.

9.1.3 Combined WID and D2D analysis

After computing the two components of path delay variation, $D_{p,D2D}(P_{D2D})$ and $D_{p,WID}(\Delta P_{WID,i})$, we compute the distribution of the total path delay D_p. Since P_{D2D} and $\Delta P_{WID,i}$ are independent random variables, this involves the convolution of the two distributions. However, since $D_{p,D2D}$ is not normal, the convolution can not be performed analytically, and must be done by discretizing the two distributions and then taking their convolution numerically. By repeating the same procedure used for the computation of D_p, we find the total path delays $D_p^{'}$, $D_p^{''}$, $D_p^{'''}$ for all possible architectures, verify them with Monte Carlo simulation, and store them in tables. A Monte Carlo verification sample is shown in Chapter 10.

9.2 Incorporating leakage, power, and area

To compute the leakage current I_{leak} and the static/dynamic power P_{total}, along with their associated variation, we use a first-order approximation similar to that used for the delay, as explained in the previous section. We calculate the I_{leak} and P_{total} for each of the transistors on the critical path considering the detailed discussion in Chapter 7. We verify these values using the leakage current and static/dynamic power formulas presented in Section 7.6. In Chapter 10, we will present several leakage and power related simulation results that validate our findings.

For an area estimate, we use CAD tools (e.g. Virtuoso Layout Editing, Cadence Inc.) to draw custom design layouts for each of the logic components used in the sub-circuits of the SRAM—such as a 6T-cell, flip-flop, mux, among others. Then we extrapolate the layout measurement to obtain an area estimate for a given component. For example, with the layout of a 6T-cell (and its associated routing in hand), we can compute the area of an entire 6T-array by multiplying the area of one 6T-cell (and its associated routing) by the number of SRAM memory cells.

9.3 Model assumptions and implementation

Although labor-intensive (mainly during the data collection for the sensitivities), the construction of a hybrid analytical-empirical model such as this one takes a reasonable time on a small cluster (weeks, not months). The initial expensive sensitivity analysis is compensated for by the time savings in the subsequent short run-times. While Hspice Monte-Carlo simulations for each of the many possible configurations of an actual large SRAM circuit can take days

(which makes such alternatives comparatively quite expensive), VAR-TX carries out the same analysis in minutes. Despite the time savings, for the circuits we have chosen, our model produces delay estimates within 8% of Hspice results. A total independent variation of 8.98% for the WID sigma of V_{th}, 4.84% for L, and 2% for V_{dd} were assumed for our variability analysis of a 16-nm node. For D2D independent variance, we assumed 4.01% for either V_{th} or L, and 2% for V_{dd}. We chose these percentages based on the manufacturing process variation forecast of ITRS [1]. Our simulations are based on ASU Predictive Technology Models (PTM) [33]. Sixty different transistor models, each with a different value for V_{TH0}, were used to model V_{th} variations for our SRAM circuits. To model gate length variations, we stipulated 20 different values of deviation from the standard minimum-size transistor length. Finally, we modeled V_{dd} variations using two extreme cases: the default supply voltage plus 1-sigma and the default V_{dd} minus 1-sigma. Every transistor in the netlist was subject to both random and spatially-correlated systematic fluctuations of V_{th}, L, and V_{dd}. The proposed model assumptions are verified through Monte Carlo simulation and validated through comparison with VARIUS [169], which show that the proposed approach produces very accurate results.

9.4 Model optimization

In addition to computing the access-time of a given SRAM system, VAR-TX performs exhaustive computations and comparisons based on user-defined parameters (i.e. SRAM size, word-size) to provide the minimum-access-time architecture/organization that satisfies desired power and area requirements from the modeled alternatives. VAR-TX, using its embedded

library of lookup tables (constructed from the linearized device delays for different configurations), does this within thirty seconds, even for large SRAM circuits with nearly countless critical parameter fluctuations. VAR-TX also provides a measure of the expected variability in this minimum access-time.

9.5 How to use the model

In addition to the proposed comprehensive modeling methodology, presented in this chapter, and the rather shorter version of the proposed comprehensive modeling methodology, discussed in our ISQED–2012 published paper (presented in Appendix A), the proposed software VAR-TX may be obtained upon request by emailing the author: jeffsrad@soe.ucsc.edu. The user can run the program by entering a desired set of four SRAM specifications, provided and explained in more detail by the software. These specifications include SRAM size and shape, number of columns, word-size, and technology node. VAR-TX will return detailed tabulated data that suggests an optimized architecture for the greatest yield, an estimated prediction of the associated access-time $T_{ACCESS-TIME}$, and the variation of access-time $\delta_{T_{ACCESS-TIME}}$, all within 30 seconds. The extended version of the model will add two additional user entries (namely the desired total power and the desired total area, acting as two constraints) in future work.

Part VI

Experimental Results

Chapter 10

Simulation Results and Analysis

The results of this thesis are based on about 2000 short, medium, and large simulations. Half the simulations were aimed at gathering the empirical data needed to generate the informative data, tables, and plots used to build our proposed model for access-time and access-time variations (presented in Chapter 9) and to help analyze the results (presented in this chapter) produced by our modeling methodology. The other half of the simulations were dedicated to producing the corresponding plots, results, and analysis for leakage power, dynamic power, energy consumption, area, and the variations of these attributes. We presented some of our power-related plots and analysis in Part III (when we discussed , among others, the power-related design challenges), and will present more of them here in Chapter 10.

For small- and medium-sized SRAMs, we ran complete simulations over two-hour to two-day periods. For 64KB and larger SRAMs, both restricted simulations and VAR-TX-result extrapolations were performed and compared to each other. In restricted simulations, only selected stages or selected critical paths are targeted (rather than the entire circuit or all the

paths). In VAR-TX extrapolations, the model was instructed to compute the critical path using its library of empirical data.

For our simulations, we have used the mixed-signal Ultrasim simulator tool (MMSIM72-Ultrasim64, Cadence Inc.). For our layout samples, we have used several other CAD tools—such as Virtuoso Schematic Editing, Virtuoso Layout Editing, and Virtuoso Analog Design Environment—to do custom designs for the various components of an SRAM circuit to help obtain a more accurate estimate of area. For our transistor modeling, we have used the Arizona State University Predictive Technology Models (ASU-PTM) [33]. Therefore, our access-time, power, and area predications are valid only insofar as the Ultrasim, CAD tool equations, and ASU-PTM transistor models are accurate.

Some other assumptions that we made to generate the plots and their accompanying analysis presented in this chapter were explained earlier in Sections 9.1 and 9.2.

We begin this chapter with verification of our proposed model through Monte-Carlo simulation (Section 10.1). We then present the validation of our model optimization in Section 10.2. In Section 10.3, we examine the behavior of the delay and delay variation for differently sized 16-nm SRAMs, starting with an access-time analysis in Section 10.3.1. We analyze the impact of the cumulative and individual V_{th}, L, and V_{dd} variability, wordline vs. bitline variability, bank variability, FMAX mean variability, area, and temperature variability on access-time in Sections 10.3.2 through 10.3.8, in the order listed. In Section 10.4, we examine the behavior of the leakage power, dynamic power, and energy consumption and their variation for differently sized 16-nm SRAMs, starting with an overview in Section 10.4.1. We analyze the impact of parameter variation on leakage current and the impact of transistor threshold voltage

(V_{th}) and temperature (T) on leakage power in Sections 10.4.2 and 10.4.3, respectively. In Section 10.4.4, we will look at some simulation results for power, leakage, and energy and discuss the probability distribution of the total power in Section 10.4.5. We conclude this chapter by predicting the yield in SRAM using our yield-estimation model (Section 10.5).

10.1 Verification by Monte-Carlo

To validate the accuracy of the approach discussed in Chapter 9, we compare the distribution of D_p (Equation (9.5)), computed through the proposed analytical approach (Section 9.1.3), with that obtained through Monte Carlo simulation. For each transistor, we model each gate parameter (the same parameter of all transistors in a gate such as inverter, NAND, etc.) as:

$$P = P_0 + \Delta P = P_0 + \Delta P_{D2D} + \Delta P_{WID} \tag{10.1}$$

where P_0 is the nominal value representing V_{th0}, L_0, or V_{dd0}. For each of the gate parameter variations ΔP (e.g., $\Delta V_{\text{th k}}$, ΔL_k, and $\Delta V_{\text{dd k}}$), there are D2D and WID components ΔP_{D2D} and ΔP_{WID}, respectively. For D2D variations, we generate a random variable for each parameter for every chip according to a normal distribution. For WID variations, there are both correlated (systematic) and random components. To capture this effect, we use the same multi-level quad-tree method discussed in Section 9.1.2 and shown in Figure 9.2 that is repeated here in Figure 10.1 for convenience.

For each quadrant, we generate a random variable according to a normal distribution.

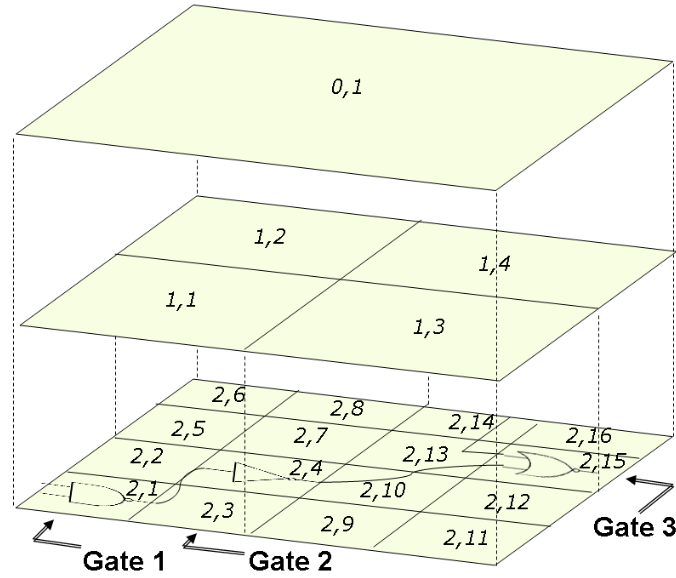


Figure 10.1: Spatial correlation modeling for WID variations (Based on Fig.1 of Agarwal [4]) (repeated for convenience).

The ΔL_{WID}, $\Delta V_{\text{dd WID}}$, and $\Delta V_{\text{th WID, sys}}$ of a transistor can be obtained by adding up all the random variables of the quadrants that contain V_{th}, L, or V_{dd}. $\Delta V_{\text{th WID, rand}}$ is obtained from a random variable which obeys a normal distribution that represents the random-WID component of V_{th}. $\Delta V_{\text{th WID}}$ is obtained by adding up $\Delta V_{\text{th WID, sys}}$ and $\Delta V_{\text{th WID, rand}}$. Similar to Section 9.1.2, transistors of Gate1 and Gate2 have strong correlation because they share the variable $rand_{0;1}$ and $rand_{1;1}$, while transistors in Gate1 have less correlation with transistors in Gate3, because they only share the variable $rand_{0;1}$. The summation ΔP_{D2D}+ΔP_{WID} for each parameter (assigned to each device separately) gives ΔP for every parameter of each device. We use Monte-Carlo simulation to generate all the random variables necessary in the model and to generate ΔP_{D2D} and ΔP_{WID} for the gate threshold voltage, length, and supply voltage for each device on the delay path in 16-nm, 45-nm, and 180-nm 64KB 6T-SRAMs. For gate parameter modeling, we apply 6 levels of quadrants (sufficient partitioning for our first order

analysis) with the top quadrant the entire SRAM and the bottom quadrant the devices. We then use the fitting curve introduced in Section 9.1.2 to obtain the coefficients a_k, b_k, and c_k of the combined WID+D2D gate delay changes ΔP to calculate the change-in-delay of every gate k on the critical path (ΔD_{P_k}). Subsequently, we can compute the delay of all possible paths D_{path} in the SRAM.

$$\begin{aligned} D_{path} &= D_0 + \Delta D_{\text{Vth i}} + \ldots + \Delta D_{\text{Vth n}} \\ +\Delta D_{\text{L i}} + \ldots + \Delta D_{\text{L n}} &+ \Delta D_{\text{Vdd i}} + \ldots + \Delta D_{\text{Vdd n}} \end{aligned} \tag{10.2}$$

$$\Delta D_{\text{Vth k}} = \partial D / \partial V_{\text{th k}} \times \Delta V_{\text{th k}} = a_k \times \Delta V_{\text{th k}} \tag{10.3}$$

$$\Delta D_{L_k} = \partial D / \partial L_k \times \Delta L_k = b_k \times \Delta L_k \tag{10.4}$$

$$\Delta D_{\text{Vdd k}} = \partial D / \partial V_{\text{dd k}} \times \Delta V_{\text{dd k}} = c_k \times \Delta V_{\text{dd k}} \tag{10.5}$$

The delay for a critical path D_{path} (Equation (10.2)) is the summation of the path nominal delay (D_0) and the additional delay caused by parameter fluctuation of each device on the path, assuming n total devices. Using this method, we simulate 2000 chips, which we find is sufficient for our statistical analysis. As an example, the plots shown in Figure 10.2 show a close match between our proposed hybrid analytical-empirical approach and the Monte Carlo simulations.

There are two additional observations that can be made from Figure 10.2.

1. The distribution of the access-time due to the cumulative WID fluctuation (dashed red curve, Figure 10.2) is almost identical to the distribution due to the combined cumulative

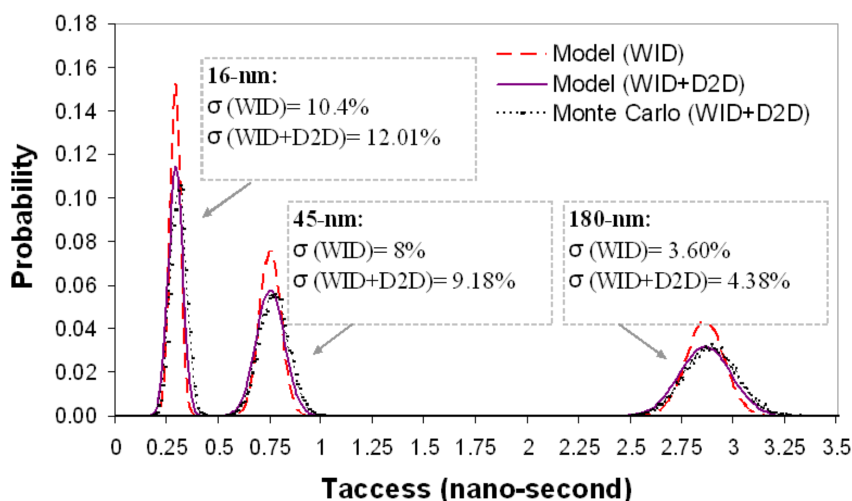


Figure 10.2: Verifying our proposed model with Monte Carlo.

WID+D2D fluctuation (solid violet curve, Figure 10.2). Therefore, we can conclude that most of the access-time variation is due to the WID variation

2. The much higher 3-sigma deviation of the delay curve of the 16-nm node suggests that a generation of performance gain could be lost in the upcoming 16-nm technology node unless new process manufacturing innovations and new circuit design methodologies are investigated and employed.

10.2 Validation of model optimization

To quantify the access-time improvement of our proposed approach, we compare the probability density function (PDF) of our optimal access-time $T_{arc,op}$ with both the PDF of our worst access-time $T_{arc,wo}$ and the PDF used in VARIUS [169], $T_{var,access}$—building on the work of Mukhopadhyay [115] that uses $T_{var,access} \propto (1/I_{dsatT1})$—for a given 45-nm 64KB 6T-SRAM

(Figure 10.3). The mean and variance used in VARIUS [169] are very similar to the mean and variance of our worst case scenario ($T_{arc,wo}$) and both are considerably different from our calculated best case scenario, which clearly confirms the optimization capability of VAR-TX.

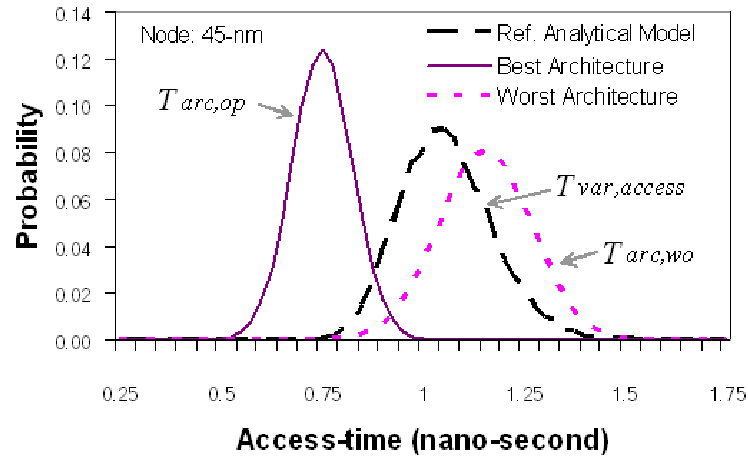


Figure 10.3: Validating optimization capability of our model.

Figure 10.4 shows the same comparison, but instead of PDF curves, we have plotted the cumulative distribution functions (CDF) of the three curves ($T_{arc,op}$, $T_{arc,wo}$, and $T_{var,access}$) shown in Figure 10.3. These three CDF curves are meant to represent the same aforementioned three cases except that they are assumed to be running at the same frequency (same mean) but different standard deviation so that we can compare them to each other. A CDF at any time shows the probability of an event occurring at or before that time. The probability error P_E can be derived from the CDF graph by looking at how far below 100% the CDF is at that point. As an example, at the 90% point, the P_E of the optimum architecture is shown in light blue in Figure 10.4. This P_E is significantly smaller than the other two cases.

Figures 10.3 and 10.4 illustrate that, by choosing the optimum architecture in an

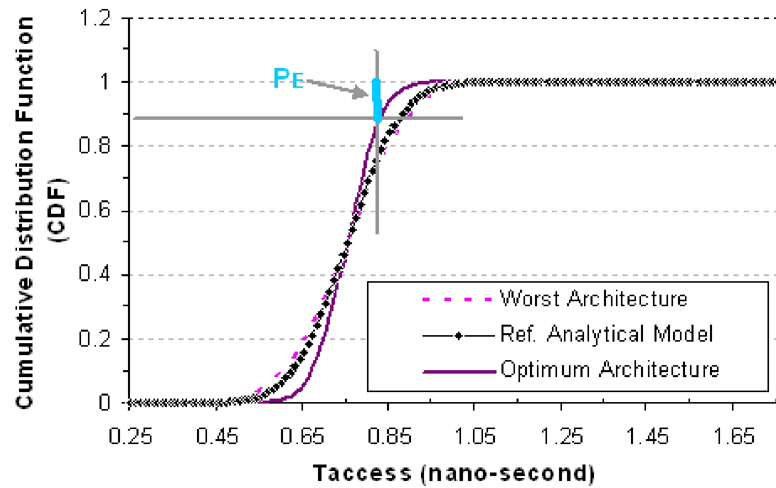


Figure 10.4: Comparing the improved cumulative distribution function (CDF) of optimum-architecture Access-Time with its counterpart CDF of VARIUS [169] and the worst-architecture Access-Time, respectively, for a given 45-nm 64KB 6T-SRAM.

SRAM design, the access-time (and therefore the yield) can be improved by up to 31% with respect to prior models such as those proposed by Mukhopadhyay and VARIUS [115, 169].

Table 10.1 compares the access-time mean, sigma, and 3-sigma of the optimum, worst, and three other architectures that fall between the optimum and worst to the access-time obtained using the Ref. VARIUS [169]. The drastic difference between the mean, sigma, and 3-sigma of the worst and optimum cases clearly emphasize the crucial role of selecting an optimum architecture in frequency improvement.

Table 10.1: Comparison of different architectures with Ref. (VARIUS [169]).

Architecture/ Design selection	No. of gates in path	Mean access-time		Standard deviation		3-sigma delay	
		ns	% imp	ns	% imp	ns	%imp
Arc 1	25	0.32	24%	0.038	30%	0.435	25%
Arc 2	29	0.39	7%	0.047	14%	0.530	9%
Arc 3	33	0.48	-14%	0.059	-8%	0.657	-13%
Ref.	(?)	0.42	0%	0.055	0%	0.584	0%
Arc Worst	38	0.54	-29%	0.065	-19%	0.735	-26%
Arc Optim	27	0.29	31%	0.035	36%	0.394	32%

10.3 Delay Simulation Results and Analysis

10.3.1 Access-time

We define *access-time* as the difference between the time addresses enter the global address bus and the time outputs are placed on the global output bus. We assume the *cycle-time* is equal to the clock cycle. Since cycle-time and its variability can be estimated readily from access-time and access-time variability, the plots presented here do not explicitly show cycle-time.

Access-time is usually somewhat smaller than cycle-time except in pipelined or post-charge circuits, where the reverse may hold true. We choose a conventional structure with an access-time about 15–20% less than the cycle-time—based on the fact that the outputs are placed on the output bus about 15–20% earlier than the clock rise of the next cycle.

We characterize our access-time results using the following five terms:

- ⊙ *ACS*, **ACcess-time Squared:**

  minimum access-time for an SRAM where the optimal organization takes a square shape.

ACS is always larger than or equal to *ACI*.

⊙ ***ACI*, ACcess-time Ideal:**

similar to *ACS*, but the optimal organization need not take a square shape.

⊙ AC_{avg}, **ACcess-time average:**

the mean access-time for an SRAM of any shape, as affected by the process variations.

⊙ ***ACH*, ACcess-time High:**

the slowest possible access-time for an SRAM of any shape, as affected by the process variations.

⊙ ***ACL*, ACcess-time Low:**

the fastest possible access-time, and the opposite of *ACH*.

The two upper curves in Figure 10.5 show two access-time traces for the 16-nm technology. The trace with the sharp peak depicts *ACS* (upper dashed line); the more linear trace just below *ACS* shows *ACI*. The lower traces in the plot analytically break *ACI* down into its components, such as bank select or precharge time. The large red diamonds surrounding *ACS* are Hspice results. The fact that our VAR-TX results deviate no more than 8% from the Hspice results strongly suggests that our model can provide equally valid access-time predictions while being much quicker than Hspice. The number triads listed in Figure 10.5 (e.g., $\frac{8:64:128}{16:2:8}$) represent number of columns(8):word-size(64):number of rows(128) in the upper sets, and total number of banks(16=2×8):number of columns of banks(2):number of rows of banks(8), in the lower sets.

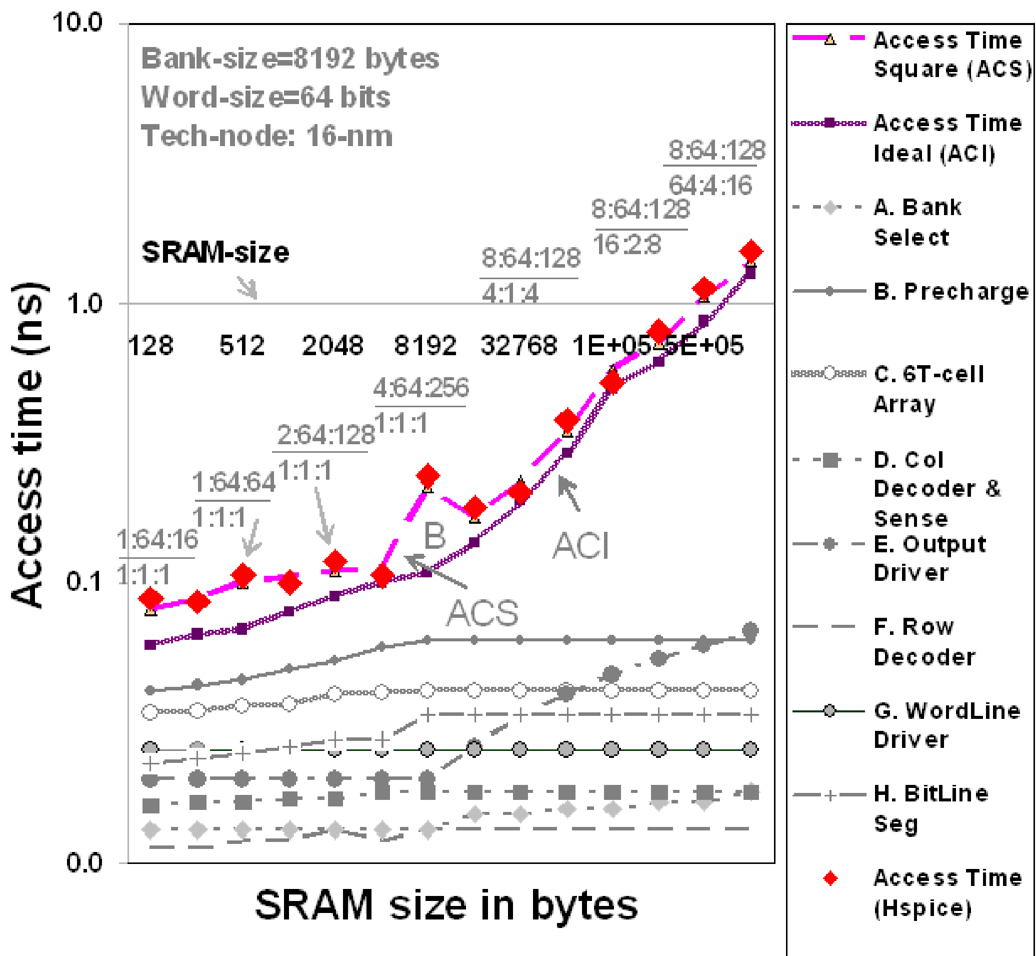


Figure 10.5: Access-time for "square" SRAM (*ACS*), Access-time for "non-square" SRAM (*ACI*), and *ACI* break-down traces.

Several observations follow from Figure 10.5:

Size and organization

SRAM delay is a function of SRAM size and organization.

Arguing square policy

Comparison of the *ACS* and *ACI* traces reveals that perfectly square SRAMs do not always produce minimum delays, especially for medium-size units. This finding contradicts

previously published work [189, 141, 167] and common expectations that have found that the minimum delays are always produced by perfectly square SRAMs (and never by non-square SRAMs). However, our model shows that it is possible, in some cases, that the delay of a non-square SRAM can be shorter than the delay of a perfectly square SRAM of the same size. This is due to the fact that the previous studies base their assumptions heavily on the Elmore delay (delay due to the resistance and capacitance of wiring/routing), which is minimized with a square shape SRAM. Although intuitively correct, those studies do not take into account the cumulative effect of differently sized components of the SRAM. For example, the longer routing delay of non-square SRAM can be compensated for by making one of the SRAM components, such as the output driver, a bit bigger while making the row decoder a bit smaller.

This means it is possible to achieve a more desirable access-time by selecting an optimum organization and architecture for the SRAM. If one compares the left side of the *ACS* and *ACI* traces in Figure 10.5, it is apparent that the SRAM access-time can be reduced by up to 31% by favoring one or more SRAM input specifications over others. For example, word-size can be favored over the number of rows. This "favoritism" involves a negligible amount of extra area and cost for more sense-amps and flip-flops.

Varius component delays

Precharge and SixTXArray component delays are much larger than the other component delays. Mitigating this is the fact that SRAM stability increases with sufficiently large pre-charging and discharging times. The large delay times for the Precharge and SixTXArray components effectively outweigh delays from the row decoder, column decoder, and wordline and bitline segmenting. SRAM delay variability tends to be partially obscured as well; this

effect will be explained further below.

ACS approaching *ACI* due to banking

Whereas the left wings of the *ACS* and *ACI* traces differ for SRAM sizes up to 8KB (Point B), the right wings are nearly linear and almost overlap. Beyond Point B, the advantage of multiple banks kicks in, changing not only output driver behavior, but also permitting bank arrangements that make *ACS* almost the equal of *ACI*. The nearly linear increase in both *ACS* and *ACI* beyond Point B is mostly due to the fact that the output driver changes from parallel to serial mode.

Benefits of intelligent balancing

Finally, we see that intelligent balancing of different SRAM components can provide two benefits. First, optimum delays for larger SRAMs can be reduced nearly to those for small- and medium-sized SRAMs. And second, when large-SRAM delays increase with SRAM size in a near-linear fashion, delay and variability become more predictable.

10.3.2 Cumulative V_{th}, L, and V_{dd} Variability

Figure 10.6 and Table 10.2 compare the 3-sigma and 1-sigma corner points of ACI, respectively, to show how the 180-nm, 45-nm and 16-nm cumulative parameter fluctuations impact access-time. Note the AC_{avg} traces, located just above the heavy *ACI* traces, for all three nodes ($AC_{avg_{180}}$, $AC_{avg_{45}}$, and $AC_{avg_{16}}$) in Figure 10.6. These AC_{avg} traces show that cumulative systematic and random variations in V_{th}, L, and V_{dd} generally increase access-time for all SRAM sizes. This means the upper corner-point variations in access-time, represented by *ACH*, are always larger than the lower corner-point variations in access-time, represented by *ACL*.

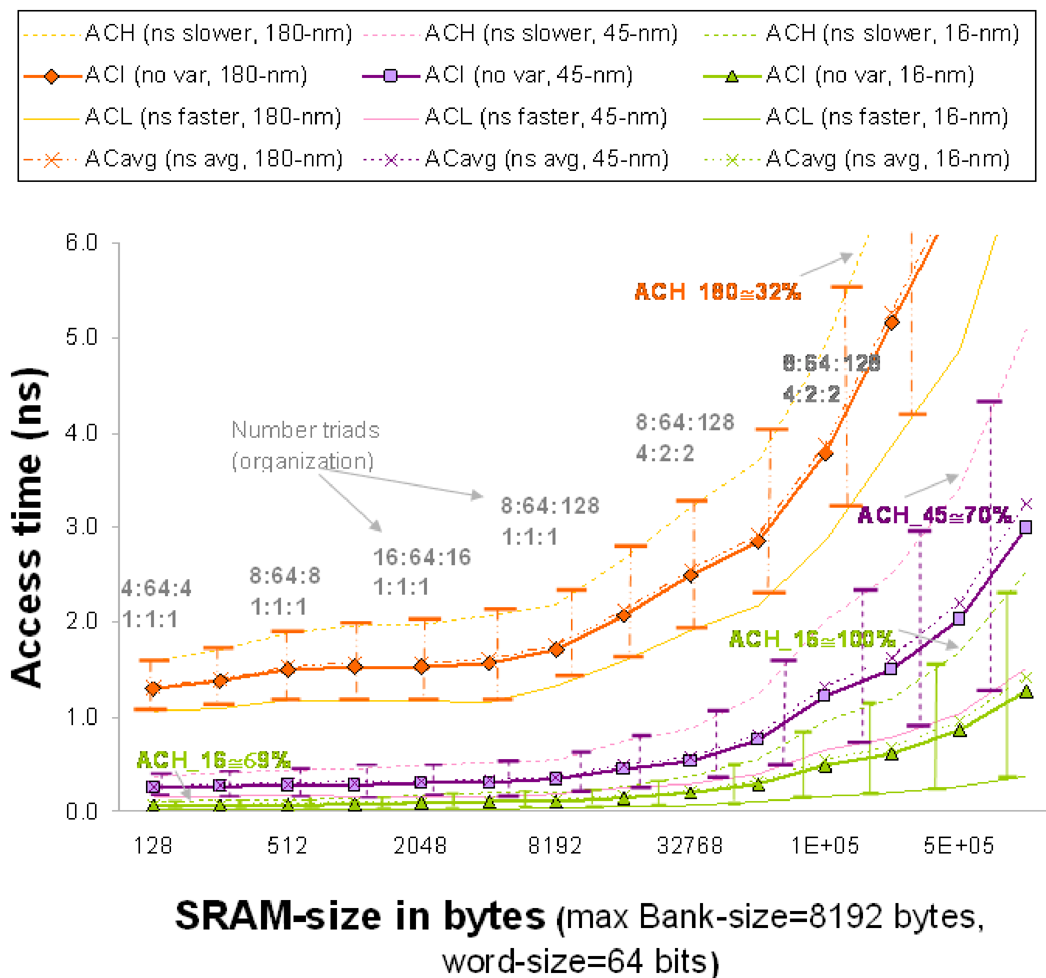


Figure 10.6: Comparing the *ACI* (ideal access-time) 3-sigma corner points (incurred due to cumulative parameter fluctuations) of 16-nm with those of 180-nm and 45-nm.

In general, the deviations of the 16-nm *ACH* and *ACL* from *ACI* are considerably larger than their 45-nm and dramatically larger than their 180-nm counterparts for any given SRAM size. For example, whereas the 180-nm *ACH* variation and 45-nm *ACH* variation reach 32% and 70% respectively, the 16-nm *ACH* variation jumps to about 100%. This alarming result highlights the need for better wafer/die manufacturing methods and design strategies to

avoid low yields and functional failures, respectively.

Table 10.2: Comparing the *ACI* (ideal access-time) 1-sigma of 16-nm (incurred due to **cumulative** parameter fluctuations) with those of 180-nm and 45-nm for different SRAM-sizes.

SRAM size (bytes)	Organization (*ACI*)	Access-Time (*ACI*) 1-sigma (in %)		
		180-nm	**45-nm**	**16-nm**
128	$\frac{4:64:4}{1:1:1}$	3.37	7.05	9.20
256	$\frac{8:64:4}{1:1:1}$	3.73	7.81	10.19
512	$\frac{8:64:8}{1:1:1}$	3.82	7.98	10.42
1024	$\frac{16:64:8}{1:1:1}$	4.24	8.87	11.57
2048	$\frac{16:64:16}{1:1:1}$	4.32	9.04	11.80
4096	$\frac{32:64:16}{1:1:1}$	4.83	10.11	13.19
8192	$\frac{8:64:128}{1:1:1}$	4.15	8.68	11.33
16384	$\frac{8:64:128}{2:1:2}$	4.23	8.85	11.55
32768	$\frac{8:64:128}{4:2:2}$	4.32	9.03	11.78
65536	$\frac{8:64:128}{8:2:4}$	4.40	9.20	12.01
131072	$\frac{8:64:128}{16:4:4}$	4.48	9.37	12.24
262144	$\frac{8:64:128}{32:4:8}$	4.57	9.55	12.47
524288	$\frac{8:64:128}{64:8:8}$	4.65	9.72	12.69
1048576	$\frac{8:64:128}{128:8:16}$	4.73	9.90	12.92

As explained in Chapters 5 and Part III (Memory Cell Operation and Design Challenges), access-time is influenced by I_{dsat} and bitline capacitance. I_{dsat}, in turn, is influenced by the process and operation parameters (V_{th}, L, and V_{dd}), and by their variation as well. In Chapter 5, we saw that I_{dsat} has a quadratic dependence on the difference between the gate voltage and threshold voltage (Vgs-V_{th}) and a linear dependence on the channel length dimensions (W and L), oxide thickness (t_{ox}), and bitline capacitance. Figure 10.7 shows the cumulative probability distribution of the access-time for four different SRAM sizes, with the assumed parameter variations presented in Section 9.2 (i.e. independent WID variations of 8.98% for V_{th}, 4.84% for L, and 2% for V_{dd} and independent D2D variations of 4.01% for V_{th} and L, and 2% for V_{dd}).

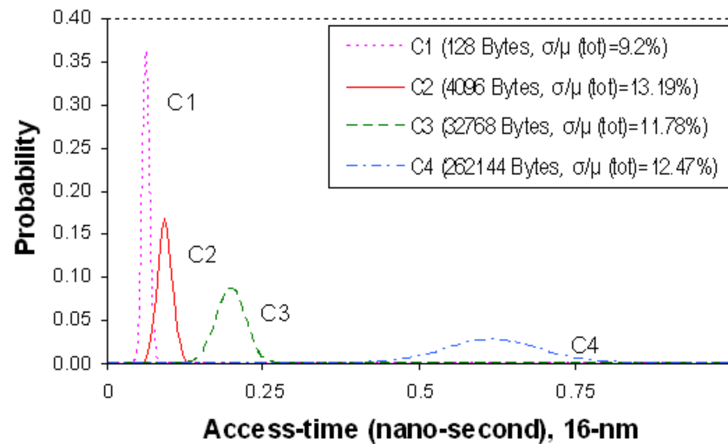


Figure 10.7: Cumulative distribution of access-time for 4 different SRAM sizes in 16-nm node.

Several observations follow from Figure 10.7:

1. SRAM access-time variation is a function of SRAM size and organization.

2. The results show that the cumulative probability of access-time variation grows with an increase in row size. The lowest row width (curve C1, 4×64, showing the smallest σ=9.2%) displays a much lower cumulative probability than the largest row width (curve C2, 32×64, showing the largest σ=13.19%).

3. Comparing the different SRAM sizes, we can deduce that as area is increased, the cumulative probability of variation is only slightly larger. This holds not just for access-time but for power as well, as will be explained in Section 10.4.5 (Probability Distribution of Total Power).

4. Comparing the PDF traces of access-time of 16-nm SRAMs with 45-nm and 180-nm SRAMs reveals that the variation of SRAM increases with technology scaling, as was

shown earlier in Figure 10.2.

10.3.3 Individual V_{th}, L, & V_{dd} Variations

Two interesting observations follow from Table 10.3, which lists the individual impacts of transistor threshold voltage, transistor length, and supply voltage variations on the access-time variation for three different technology generations and several different SRAM sizes:

1. Variation of the access-time due to V_{th} variation is much larger for the newer nodes than for the older nodes. For example, whereas the variation of access-time due to V_{th} variation barely reaches 2.5% in 180-nm 64KB, it easily exceeds 7.5% and 9.8% in 45-nm 64KB and 16-nm 64KB, respectively. A similar trend in the 32-nm node is observed elsewhere [169].

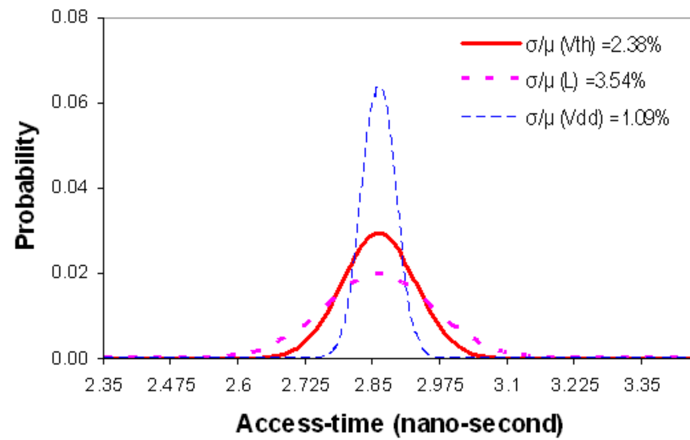
2. Scaling the technology from the older node (180-nm) to the newer nodes (16-nm and/or 45-nm) shifts the main contribution to the variation in access-time from L variation (about 3.6% in 180-nm 64KB) to V_{th} variation (about 9.83% in 16-nm 64KB). Scaling from the larger to the smaller technology impacts V_{th} variation drastically, while the variation of access-time due to V_{dd} variation increases a modest 5% or so. The change in the relative impact of parameter variation between the technology nodes is particular to each parameter, of course. Oxide thickness reduction accounts for most of the V_{th} change; lithography improvements that allow fabrication of smaller transistors at higher precision impact L; and reducing the supply voltage from $\sim$1.8V to $\sim$1.1V and to $\sim$0.9V affects

V_{dd} variability.

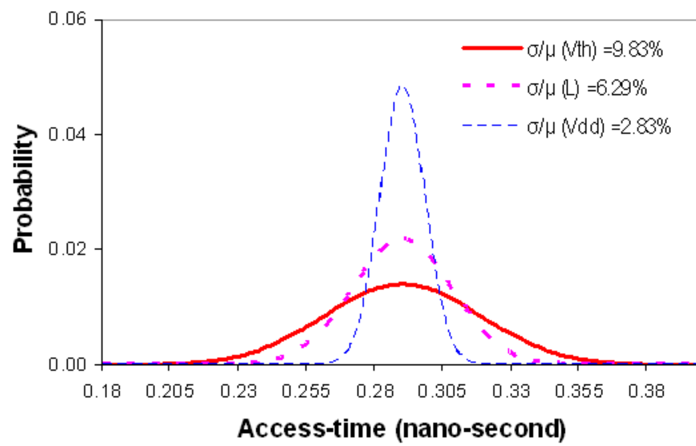
Table 10.3: Comparing the *ACI* (ideal access-time) 1-sigma of 16-nm (incurred due to **individual** parameter fluctuations) with those of 180-nm and 45-nm for different SRAM-sizes.

SRAM size (bytes)	Organization (*ACI*)	Access-Time (*ACI*) 1-sigma (in %)								
		180-nm			**45-nm**			**16-nm**		
		V_{th}	L	V_{dd}	V_{th}	L	V_{dd}	V_{th}	L	V_{dd}
128	$\frac{4:64:4}{1:1:1}$	1.82	2.71	0.84	5.78	3.68	1.65	7.53	4.82	2.17
256	$\frac{8:64:4}{1:1:1}$	2.02	3.00	0.92	6.40	4.07	1.83	8.34	5.34	2.40
512	$\frac{8:64:8}{1:1:1}$	2.06	3.07	0.95	6.54	4.16	1.87	8.53	5.46	2.45
1024	$\frac{16:64:8}{1:1:1}$	2.29	3.41	1.05	7.27	4.63	2.08	9.47	6.06	2.73
2048	$\frac{16:64:16}{1:1:1}$	2.34	3.48	1.07	7.41	4.72	2.12	9.66	6.18	2.78
4096	$\frac{32:64:16}{1:1:1}$	2.61	3.89	1.20	8.28	5.27	2.37	10.80	6.91	3.11
8192	$\frac{8:64:128}{1:1:1}$	2.24	3.34	1.03	7.11	4.53	2.04	9.27	5.93	2.67
16384	$\frac{8:64:128}{2:1:2}$	2.29	3.41	1.05	7.25	4.62	2.08	9.46	6.05	2.72
32768	$\frac{8:64:128}{4:2:2}$	2.33	3.47	1.07	7.40	4.71	2.12	9.64	6.17	2.78
65536	$\frac{8:64:128}{8:2:4}$	2.38	3.54	1.09	7.54	4.80	2.16	9.83	6.29	2.83
131072	$\frac{8:64:128}{16:4:4}$	2.43	3.61	1.11	7.68	4.89	2.20	10.02	6.41	2.88
262144	$\frac{8:64:128}{32:4:8}$	2.47	3.67	1.13	7.83	4.98	2.24	10.20	6.53	2.94
524288	$\frac{8:64:128}{64:8:8}$	2.52	3.74	1.15	7.97	5.07	2.28	10.39	6.65	2.99
1048576	$\frac{8:64:128}{128:8:16}$	2.56	3.81	1.17	8.11	5.16	2.32	10.57	6.77	3.04

In Figure 10.8(a), we show the probability density function (PDF) of 180 nm *ACI* due to WID+D2D variation for each device parameter separately (V_{th}, L, and V_{dd}). The mean of each distribution is aligned at 0.29 ns. Comparing the three PDFs with each other, it is clear that *ACI* due to V_{dd} variation has the narrowest distribution, followed by V_{th} and L. This difference in width of the three PDF curves is a direct measure of the standard deviation, and therefore variability, of *ACI* due to the three parameters. The 3-sigma delay due to each of these three parameters follows the same pattern, meaning that L causes the worst deviation in access-time for the 180 nm node.



(a) 180-nm



(b) 16-nm

Figure 10.8: Individual Distribution of Access-time for (a) 180-nm 64KB SRAM and (b) 16-nm 64KB SRAM.

Figure 10.8(b) shows a plot similar to Figure 10.8(a), except that it is for the 16 nm node and it is the PDF due to V_{th} that has the widest width. The same pattern holds for the 45-nm case (not shown) as well. This and other similar experimental results confirm that the performance limiting parameter in newer nodes such as 45-nm and 16-nm is not L, but the V_{th}. In other words, going from older nodes to newer nodes has swapped the magnitude variability

role of L and V_{th}. Table 10.3 validates our preceding discussion regarding the change in the impact of L and V_{th} variability on access-time, due to technology scale down. The effect of V_{dd} variation on access-time in newer nodes, though, does not show much change, comparatively.

10.3.4 Wordline vs. Bitline Variability

Figure 10.9 compares wordline 3σ corner-point delay variability to bitline 3σ corner-point delay variability for the 16-nm node.

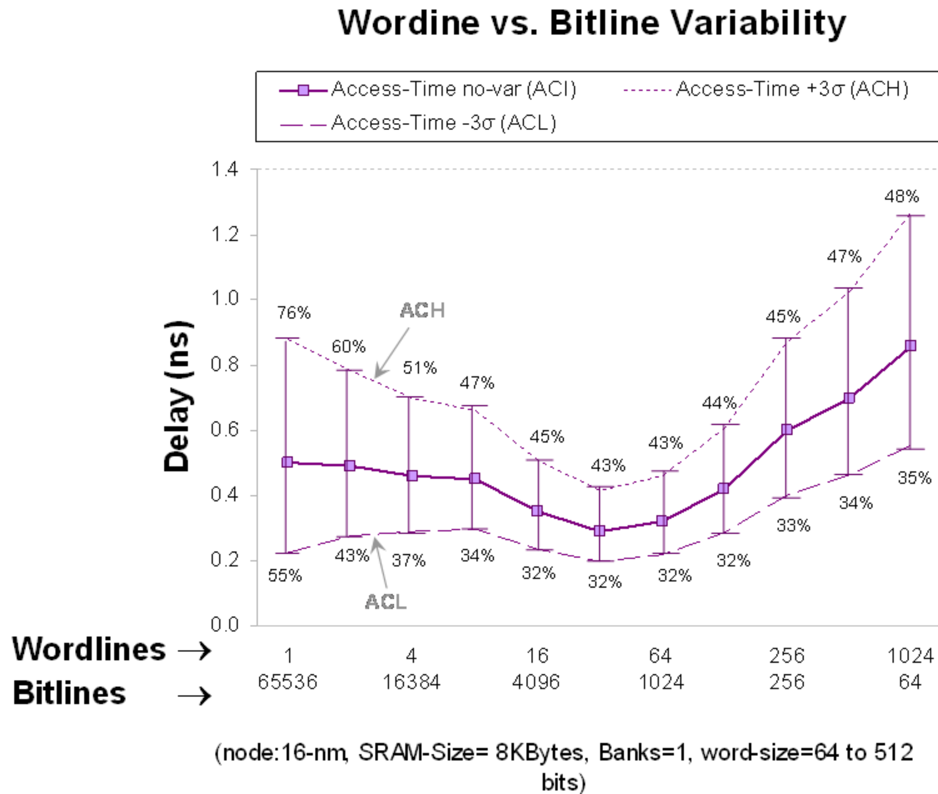


Figure 10.9: Wordline vs. Bitline 3σ corner-points (*ACH* and *ACL*) Variability of 16-nm SRAM.

To provide the possibility of comparing the upper (the slowest possible access-time) and the lower (the fastest possible access-time) 3σ corner-points to *ACI*, both *ACH* and *ACL*

traces are shown. The horizontal axis extends from minimum modeled wordlines and maximum modeled bitlines at left to the reverse case of maximum wordlines and minimum bitlines at right. Several interesting observations follow from Figure 10.9:

1. We see that delay variability in the large-bitline cases substantially exceeds delay variability in the large-wordline cases. One can control long-bitline variability to a degree, with well-chosen bitline segmenting. Control over long-wordline variability is harder to achieve since oxide thickness—and therefore V_{th}—varies across long wordlines, and is especially problematic at the extreme ends.

2. The decrease in SRAM delay variability with a larger number of wordlines comes at a price, however. When there are more than the optimal number of wordlines (more than 128 in our 6T-SRAM), access-time climbs steeply. This effect is less pronounced for the 45-nm and 180-nm nodes (not shown). The physical parameter most responsible for the variation difference between the nodes is the 50% reduction in oxide thickness in going from a larger node to a smaller node.

3. The upper 3σ variation *ACH* is always larger than its corresponding lower 3σ variation *ACL*. This means the access-time of SRAM has a higher probability of becoming slower than becoming faster, due to process variations.

4. The rise of the access-time on the extreme left-end of the *ACI* trace is due to the large number of columns (i.e. 128) used in non-optimum organizations (i.e. $\frac{128:512:1}{1:1:1}$). Similarly, the rise of the access-time on the extreme right-end of the *ACI* trace is due to the

large number of wordlines (i.e. 1024) used in some other non-optimum organizations (i.e. $\frac{1:64:1024}{1:1:1}$).

5. Both bitline and wordline variability fall to a minimum in the middle of the plot, where optimum organizations (i.e. $\frac{32:64:32}{1:1:1}$) are found. This finding further validates our VAR-TX model.

10.3.5 Bank Variability

In general, the variability of large SRAMs declines when SRAMs are divided into smaller sized banks. Such decline in variability, however, comes at the expense of increased delay, power, and area in most cases.

Figure 10.10 shows how access-time variability improves in the 16-nm 64Kb 6T-SRAM when a larger number of banks are used. Similar, but smaller, improvements are seen for the 45-nm (32% less) and 180-nm (70% less) nodes as well (not shown). There are two different sets of plots in Figure 10.10, each set composed of three traces. One set (purple traces) represents an organization with a wordwidth of 64 bits and the other set (orange traces) represents an organization with a wordwidth of 1024 bits. In both sets, the *ACI* trace represents the ideal access-time with no variability, and the +sigma and —sigma traces represent the upper variation (slower) access-time and the lower variation (faster) access-time, respectively.

Looking at Figure 10.10, we can observe that an increase in bank number from 1 to 128 leads to a decrease in *ACI* variation, as the upper variation of *ACI* ($+\sigma$ trace) decreases from 13.7% to 10.8% for BW=64 (purple) and from 16.9% to 12.8% for BW=1024 (orange). This means SRAM variability decreases as the number of banks increases. The main reason

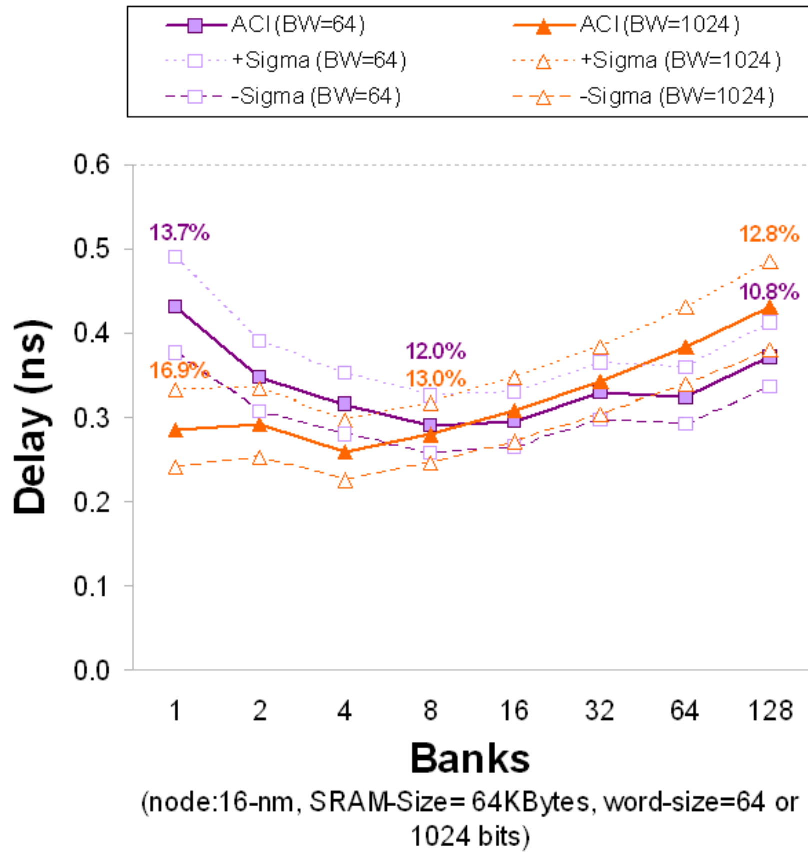


Figure 10.10: Bank Variability; Access-time variation vs. number of banks—illustrating 1-sigma corner-points (+sigma, -sigma) variability of *ACI* (ideal access-time) for two different organizations (word-width=64bits and wordwidth=1024bits) for a 16-nm 64KB SRAM divided into 1 to 128 banks.

SRAM variation ($\pm\sigma$) declines with a larger number of banks is related to the smaller number of bitlines used in smaller banks. A smaller number of bitlines means shorter wordlines. This, in turn, means more correlation and a smaller possibility of a mismatch between the 6T-cells on the same short wordlines, and also a smaller probability of variation between the transistors in those cells. Other factors such as a smaller number of rows, smaller area, and smaller loading effects on the output bus used in smaller sized banks are also among the reasons why SRAM

variation ($\pm\sigma$) declines with a larger number of banks. However, the impact of these secondary factors on variability is not as much as that incurred by the bitlines. Our model's distance correlation incorporates the increase in the probability of oxide thickness variation as wordlines and bitlines increase in length.

The price for reducing overall SRAM variation by raising the number of banks (above the optimum number of banks) is a considerable area increase, a tolerable power increase, and a delay increase due to output bank loading. This trade off phenomenon is shown in Figure 10.10 and is illustrated by PDF curves in Figure 10.11(a) and 10.11(b). Figure 10.10 shows that the access-time for both cases starts and/or continues to increase as the 64kB SRAM is divided into more than 8 banks. Similarly, Figure 10.11(a) and 10.11(b) show that the access-time (or the mean) of the curves x_{128} and y_{128} (having smaller sigma) is larger than the curves x_8 and y_8 (having moderately larger variation). Simply put, the price for $\sim$3% improvement in variability is about 28% decline in speed.

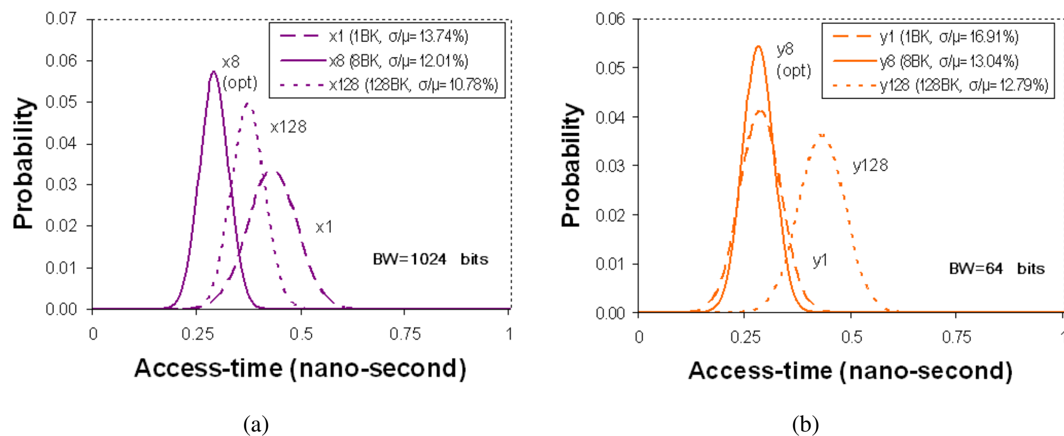


Figure 10.11: Bank Variability; illustrating the distribution of *ACI* (ideal access-time) for two different organizations—(a) BW=64 bits and (b) BW=1024 bits—for a 16-nm 64KB SRAM divided into 1 to 128 banks.

Designers should also be aware of potential SRAM yield declines when the number of banks increases. The latter effect stems from the increased hardware-failure probability with larger transistor numbers.

Table 10.4 summarizes the mean and standard deviation of the distribution of access-time (PDF) for the aforementioned two different organization sets (wordwidth=64bits and word-width=1024bits), as the number of banks is swept from 1 to 128. Looking at Table 10.4, we can see that the reduction in sigma generally corresponds to an increase in mean and vice versa.

Table 10.4: Analysis of Mean and standard deviation of Ideal Access-Time (ACI) for two different organizations, one with Wordwidth=64 bits and the other with Wordwidth=1024 bits, in 16-nm SRAMs of different bank numbers.

Number of Banks	Wordwidth = **64** bits (**x** curves)			Wordwidth = **1024** bits (**y** curves)		
	Organization	Mean (ns)	Std (%)	Organization	Mean (ns)	Std (%)
1	$\frac{32:64:256}{1:1:1}$	0.43	13.7	$\frac{16:1024:32}{1:1:1}$	0.28	16.9
2	$\frac{32:64:128}{2:1:2}$	0.35	12.6	$\frac{8:1024:32}{2:1:2}$	0.29	14.6
4	$\frac{32:64:64}{4:2:2}$	0.31	12.2	$\frac{8:1024:16}{4:2:2}$	0.26	14.5
8	$\frac{32:64:32}{8:2:4}$	0.29	12.0	$\frac{4:1024:16}{8:2:4}$	0.28	13.0
16	$\frac{16:64:32}{16:4:4}$	0.29	11.3	$\frac{4:1024:8}{16:4:4}$	0.31	13.1
32	$\frac{8:64:32}{32:4:8}$	0.33	10.8	$\frac{2:1024:8}{32:4:8}$	0.34	12.21
64	$\frac{8:64:16}{64:8:8}$	0.32	11.0	$\frac{2:1024:4}{64:8:8}$	0.38	12.4
128	$\frac{4:64:16}{128:8:16}$	0.37	10.8	$\frac{2:1024:2}{128:8:16}$	0.43	12.8

Table 10.4 also shows that the variability of SRAMs using narrower wordwidth architecture (i.e. wordwidth=64 bits) is less than the variability of SRAMs using wider word-width architecture (i.e. wordwidth=1024 bits) for a same number of banks. For example, in the case of SRAM having only one bank, a sigma=13.7% for wordwidth=64 bits is less than a sigma=16.9% for wordwidth=1024 bits. Similarly, in case of SRAM having 32 banks, a sigma =10.8% for wordwidth=64 bits is less than a sigma=12.2% for wordwidth=1024 bits.

Finally, the results in Table 10.4 (as well as the comparison between the x_{128} and y_{128} traces in Figure 10.11) reveal that both the mean and the sigma of the organizations with larger wordwidth (i.e. wordwidth=1024bits) are larger than the mean and sigma of the organizations with smaller wordwidth (i.e. wordwidth=64 bits). Such a difference, however, is less of a concern since the SRAM can typically be designed around the optimum architecture specifications. For example, the designer can pick a y_8 trace specification over a y_{128} trace specification when having a wordwidth of 1024 bits is desired.

All this means that, although overall variability decreases and overall reliability increases in SRAMs with larger bank numbers, the delay times soar and the yields decline to some degree. Luckily, such an increase in delay and decline in yield is generally acceptable in the optimally designed architecture cases, but is not necessarily the case for non-optimally designed architectures. The best approach, therefore, is to design around the optimum architecture, where the access-time is close to the smallest possible delay and has a modest variation. Whether or not such a balance between delay and variation offered by optimal architecture design can be tolerated will depend on the individual application.

10.3.6 FMAX Mean Variability

Table 10.5 is a summary of the maximum-frequency mean variations of access-time of a 64kB SRAM for the three technology nodes: 16-nm, 45-nm, and 180-nm. The three variations are: die-to-die (DTD), completely random within-die (R-WID), and completely systematic within-die (S-WID). FMAX mean, die-to-die, and within-die concepts are explained in more detail by Keith Bowman [29]. Each figure in Table 10.5 is the average of 10 simulations with

a maximum deviation of 8%. Our results in this section agree with the discussion presented by Keith Bowman [29].

Table 10.5: FMAX (maximum frequency) MEAN Variability (shown in 1-sigma) for a 64KB SRAM in three different technology nodes.

Tech node	**DTD**	**R-WID**	**S-WID**
180-nm	2.5%	1.7%	3.2%
45-nm	4.5%	3.7%	7.1%
16-nm	6.0%	4.9%	9.2%

As Table 10.5 confirms, the most significant performance limiter is the fluctuation in S-WID. Unlike R-WID fluctuations (which have an averaging effect) and DTD fluctuations (which mostly affect the standard deviation of FMAX), S-WID fluctuations directly impact the FMAX mean. This higher impact of S-WID fluctuations on the FMAX mean holds for all sizes of SRAMs and in all three nodes (not shown). If one compares the drastic differences between the FMAX mean of the 16-nm node with those of the 45-nm and 180-nm nodes in Table 10.5, it is not unreasonable to anticipate that essentially a generation of performance gain could be lost due to systematic within-die fluctuations in the upcoming 16-nm technology node. This loss will likely occur unless new innovations in manufacturing process controls (e.g., stepper lens aberration, chemical-mechanical polishing (CMP), etc.), and new circuit design methodologies (such as row or column redundancy) are investigated and employed.

The results presented here are only a small fraction of the analysis that VAR-TX is capable of performing.

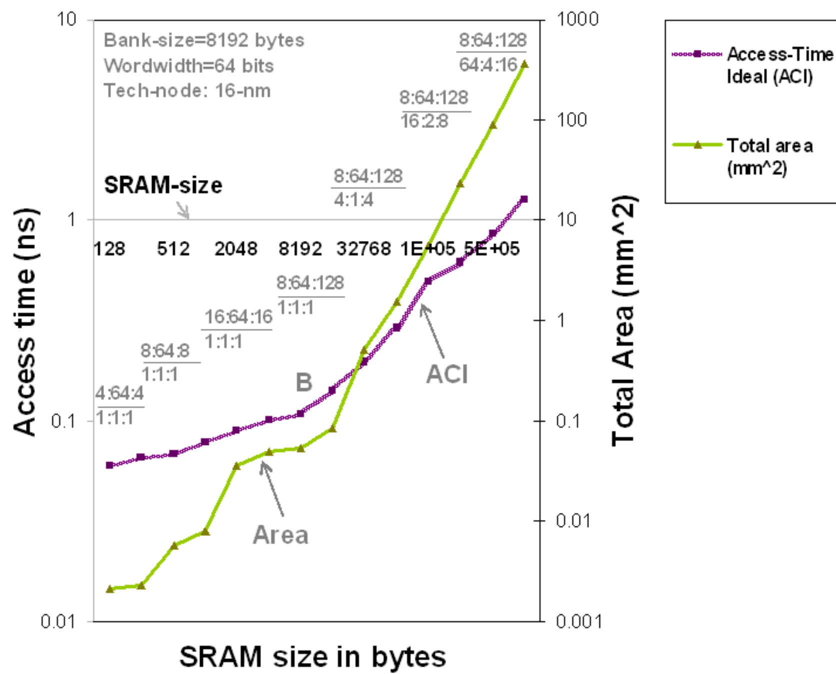


Figure 10.12: Area (right hand axis) showing higher increase rate for each doubling of SRAM sizes, as compared to that of access-time (left hand axis).

10.3.7 Area vs. SRAM size

Figure 10.12 plots the area and access-time on a logarithmic scale for different SRAM sizes. From the traces of the plot, it is reasonable to deduce that both the access-time and area of SRAM show an almost linear dependency (in a global sense) on the SRAM size, especially after point B. The area, however, increases with SRAM size more rapidly than access-time. This higher rate of increase is mainly due to the addition of bank decoders, bank-interconnects, and several banks, each having their own main components (such as row-decoder, precharge, etc.), to the SRAM design—which results in increasing the SRAM area by a factor of 3 to 4 for each doubling of the SRAM size.

10.3.8 Temperature Impact on Relative Switching Frequency

Temperature variation is caused by spatially- and temporally-varying factors. These variations are becoming more severe and harder to tolerate as technology scales to submicron feature sizes. As discussed in Section 6.1, of the three key process parameters subject to variation (V_{th}, L_{eff}, and V_{dd}), threshold voltage (V_{th}) is the most important because its variation has a substantial impact on two major properties of the SRAM/processor: the frequency it attains and the leakage power it dissipates. Moreover, V_{th} is also a strong function of temperature, which increases its variability [169].

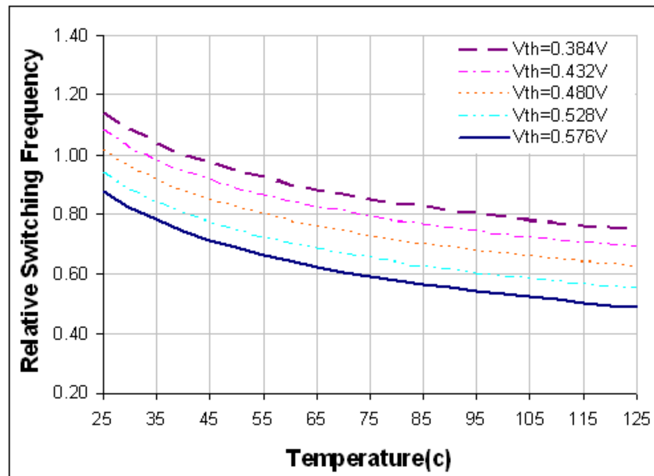


Figure 10.13: Relative switching frequency versus temperature for different threshold voltages. We use V_{th}=0.480V at 27C (room temperature) as reference.

The simulated plots shown in Figure 10.13 (for 16-nm) agree with the corresponding results in VARIUS [169], except that the declining slope of the relative switching frequency due to a temperature increase is almost twice as steep than for VARIUS (32-nm). Such a high rate in the relative switching frequency is most likely due to the existence of a larger drain current, larger wire resistance, and larger junction capacitance (C_j) inherent in the smaller 16-

nm technology node modeling.

One of the most harmful effects of variation is that some sections of the chip are slower than others—either because their transistors are intrinsically slower or because high temperature or low supply voltage renders them so. As a result, circuits in these sections may be unable to propagate signals fast enough and may suffer timing errors. To avoid these errors, designers in upcoming technology generations (i.e., 16-nm) may slow down the frequency of the processor or create overly conservative designs. It has been suggested that parameter variation may wipe out most of the potential gains provided by one technology generation [29].

As we discussed in Chapters 1 and 7, the important first step to redress this trend is to understand how parameter variation affects the timing errors in high-performance SRAMs and processors. Based on this, we can devise techniques to cope with the problem—hopefully recouping the full gains offered by every technology generation. Chapter 7 attempted to accomplish this task by presenting several recent advanced techniques that can either remedy or minimize such adverse effects on the chip performance. These techniques were then incorporated into our proposed VAR-TX modeling in Chapter 9. Here, in this section, we present two more plots that illustrate the impact of temperature on frequency. We illustrate the impact of temperature on leakage current and leakage power in Sections 10.4.2 and 10.4.4, respectively.

As discussed in Section 7.1, the impact of temperature on delay is not as dramatic as the impact of temperature on leakage power. Figure 10.13 illustrates the impact of temperature on the relative switching frequency for a 16-nm 64KB SRAM. As we can see in Figure 10.13, the dependence of temperature on the relative switching frequency is not very strong. All five plots (with the middle one V_{th}=0.220V used as reference) follow a similar modest decreasing

trend.

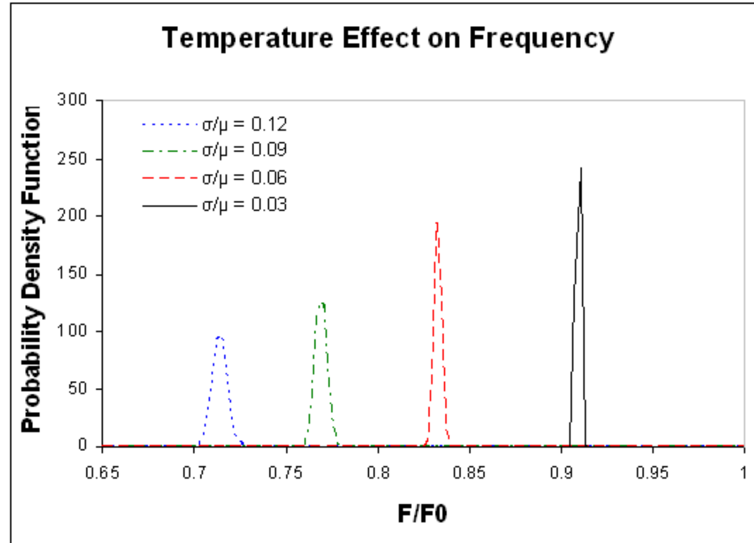


Figure 10.14: Probability distribution of the relative chip frequency as a function of V_{th}'s σ (varied due to temperature change). We use V_{th}=0.480V at 27C, 27 gates in the critical path, and 2000 critical paths in our 16nm 64KB SRAM design.

Figure 10.14 shows the effect of temperature on the frequency of SRAM. Assuming that every critical path in an SRAM consists of n_{cp} gates and that a modern SRAM chip has hundreds of critical paths, Bowman et al. [29] compute the probability distribution of the longest critical path delay in the chip (max $\{T_{cp}\}$). Such a path determines the SRAM frequency (1/maxTcp). Using this approach, we find that the value of V_{th}'s σ (resulting from a variation in temperature) affects the chip frequency.

Figure 10.14 shows the probability distribution of the chip frequency for four different values of V_{th}'s σ. A smaller σ represents a smaller variation in V_{th} due to smaller range of possible temperature changes (i.e. 27C to 50C) and a larger σ represents a larger variation in V_{th} due to larger range of possible temperature changes (i.e. 27C to 125C).

The frequency (F) is normalized to the case of an SRAM without V_{th} variation (F_0).

The PDF curves in Figure 10.14 show that as σ increases: (1) the mean chip frequency decreases and (2) the chip frequency distribution becomes more spread out. In other words, given a batch of chips, as the σ of V_{th} increases, the mean frequency of the batch decreases and at the same time, an individual chip‘s frequency deviates more from the mean.

10.4 Power Simulation Results and Analysis

10.4.1 Overview

Variations in process parameters and/or operating conditions lead to variations in sub-threshold leakage currents that are exponentially dependent on transistor threshold voltage, temperature, and supply voltage. This leads to a rapid growth in the gate leakage as the technology is scaled down.

Power and its variation has become a design constraint not only in the domain of mobile devices, but also in high performance processors housing SRAMs. Dynamic power is caused by switching activity in CMOS circuits. It is the major source of the total power dissipation in today‘s process generation. However, static power, which is due to leakage current in the quiescent state of a circuit, has gained more importance over the last few years. Technology scaling is increasing both the absolute and relative contribution of static power dissipation. According to ITRS prediction [1], in the next several processor generations, leakage may constitute a much bigger portion of the of total power dissipation as compared to today‘s processor generation.

Recently, a great deal of research in the architecture community has focused on reduc-

ing leakage power in cache and SRAM [165, 55, 112]. As we mentioned in Chapter 7, leakage control at the architecture level is attractive because it can affect large groups of circuits at once (e.g. cache lines, SRAM banks, or the entire cache/SRAM). The VAR-TX model is an example of such an architectural technique.

10.4.2 Impact of Parameter Variations on Leakage Current

Subthreshold leakage is the main source of leakage in current and future technologies, especially since the accelerated adoption of high-k gate dielectric was set to reduce gate leakage 100-fold [38].

Leakage current is influenced by the threshold voltage, physical channel dimensions, channel/surface doping profile, drain/source junction depth, gate oxide thickness, V_{dd}, temperature, and variations in all these parameters. For long channel devices, the leakage current is dominated by leakage from the drain-well and well-substrate reverse-bias pn junctions. For short channel transistors, because of the low threshold voltage, sub-threshold leakage is much higher. As gate oxides have continued to scale, gate leakage has also become important. Rabaey [141] and Keshavarzi et al. [82] give an overview of these different leakage mechanisms. The following subthreshold leakage equation—which is based on that of HotLeakage [195], itself a simplification of the full BSIM3 SPICE model—allows an explicit account of temperature, supply voltage, and threshold voltage, as well as the important DIBL (drain induced barrier lowering) effect, which is sensitive to supply voltage.

$$I_{leakage} = \mu_0 \cdot C_{ox} \cdot \frac{W}{L} \cdot e^{b(V_{\mathrm{dd}} - V_{\mathrm{dd0}})} \cdot v_t^2 \cdot \left(1 - e^{\frac{-V_{\mathrm{dd}}}{v_t}}\right) \cdot e^{\frac{-|V_{\mathrm{th}}| - V_{off}}{n \cdot v_t}} \quad (10.6)$$

where

μ_0 is the zero bias mobility,

C_{ox} is gate oxide capacitance per unit area,

W/L is the aspect ratio of the transistor,

The exponential term $e^{b(V_{\mathrm{dd}}-V_{\mathrm{dd0}})}$ is the DIBL factor. V_{dd0} is the default supply voltage for each technology (V_{dd0}=0.9 for 16-nm),

v_t = KT/q is the thermal voltage,

V_{th} is threshold voltage which is also a function of temperature,

n is the subthreshold swing coefficient,

V_{off} is an empirically determined BSIM3 parameter which is also a function of threshold voltage.

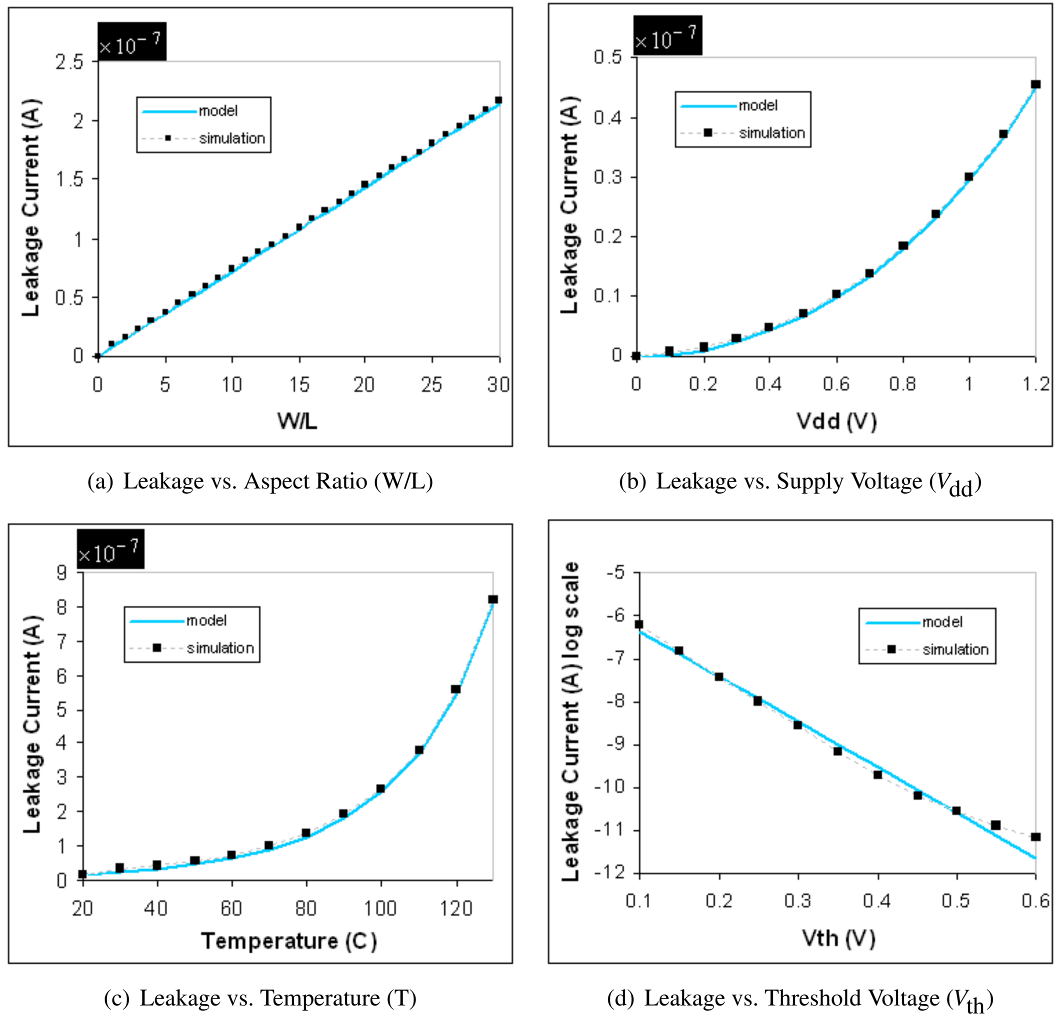
In these parameters, μ_0, C_{ox}, W/L and V_{dd0} are statically defined parameters; the DIBL coefficient b, subthreshold swing coefficient n, and V_{off} are derived from the curve fitting method based on the transistor level simulations; V_{dd}, V_{th} and $v_t = KT/q$ are calculated dynamically in the simulations.

The above equation is based on two assumptions:

1. V_{gs}=0—we only consider the leakage current when the transistor is off.

2. V_{ds}=V_{dd}—we only consider a single transistor here; the stack effect and the interaction among multiple transistors (within a cell) are taken into account by using Equation (10.7).

Figure 10.15 shows the comparison of leakage current calculated by Equation (10.6) and the transistor-level simulation. From Figure 10.15(a), (b), and (c), we can see that for the



(a) Leakage vs. Aspect Ratio (W/L)

(b) Leakage vs. Supply Voltage (V_{dd})

(c) Leakage vs. Temperature (T)

(d) Leakage vs. Threshold Voltage (V_{th})

Figure 10.15: Comparisons of the analytical model [195] against our circuit-level simulation results for 16-nm.

ratio of transistor channel width over length (W/L), supply voltage (V_{dd}), and temperature (T), the results match our simulation results. We can also observe a good match with the simulation results in Figure 10.15(d), except in the last part of the trace where the decrease of the modeled leakage current levels off. This is due to the simplicity of Equation (10.6), which only considers the subthreshold leakage and DIBL effect. This issue is only of concern if the threshold voltage

and/or its variation are beyond the normal values.

For a specific cell, the leakage current can be found through the following equation [195]:

$$I_{cell_leakage} = n_n \cdot k_n \cdot I_n + n_p \cdot k_p \cdot I_p \tag{10.7}$$

where

n_n and n_p are the numbers of NMOS and PMOS transistors in the cell,

k_n and k_p are the design factors determined by the stack effect and aspect ratio of NMOS and PMOS transistors, respectively, and they can be derived from transistor-level simulation of an actual design of a cell of interest, and

I_n and I_p are the calculated leakage current of NMOS and PMOS according to Equation (10.6). When the aspect ratio W/L=1, we call them *unit leakage*.

(The stack effect refers to the additional reduction in leakage when multiple transistors connected in series are off; for example, sleep transistors take advantage of this.) The authors in [195] show that k_n and k_p do not have explicit relations with a different technology.

Gate leakage is strongly dependent on the gate oxide thickness t_{ox} (due to the direct tunneling current) and supply voltage. It is weakly dependent on the temperature [195]. From transistor-level simulations, we derive these factors with the curve-fitting method and incorporate it into our model.

Due to both D2D and WID parameter variation, there is a significant variation in

leakage power. Thus, parameter variation must be taken into account when approximating leakage current and power. D2D variation can be characterized as a global mean and variance while WID variation can be characterized as the deviation occurring spatially within any one die.

Using a similar procedure explained in Chapter 8, we give the specific mean μ, variance σ, and the number of samples N for each of the three parameters (V_{th}, L, and V_{dd}) and generate N Gaussian distribution samples. By combining these N Gaussian distributions, we then obtain the leakage currents and leakage power distributions accordingly. The mean (μ) and sigma (σ) of those leakage currents (and leakage power) are used in the following simulations in order to show the effects of the parameter variation.

10.4.3 Statistical Estimation and Distribution of Leakage Current in SRAM

As mentioned in Chapter 7 (Section 7.6), the variation in the electrical characteristics (especially variations of V_{th}) of the transistors of a cell results in significant variation of the leakage of the cell (particularly for the sub-threshold leakage). The mean (μ_{LCELL}) and standard deviation (σ_{LCELL}) of the leakage of a cell, considering V_{th}, L, and V_{dd} fluctuation, can be obtained using a process similar to the one we have used for modeling the delay of the critical path (Chapter 9). Since the sub-threshold leakage is an exponential function of the threshold voltage, we have assumed a lognormal PDF to describe the distribution of the cell leakage [129]. Figure 10.16 shows that the lognormal distribution model, with the mean and the standard deviation estimated using the method described in Chapter 9, closely follows the Monte Carlo simulation results.

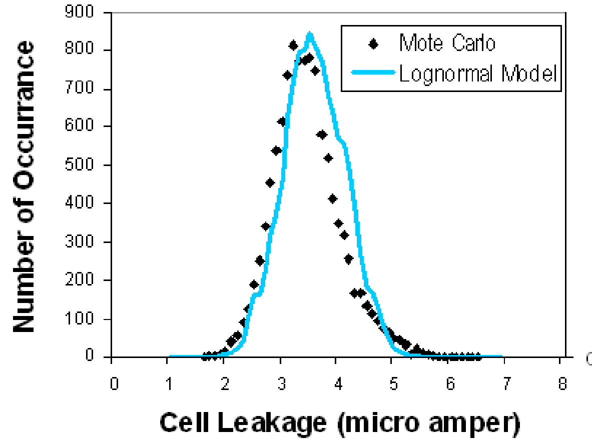


Figure 10.16: Distribution leakage of a 16-nm SRAM cell (I_{leak}).

If we assume that the different cells are independent and identically distributed random variables (i.e., the mean and the standard deviation of the leakage of all the cells are same), we can compute the estimated total SRAM array leakage through the following equation:

$$I_{LeakMem} = \sum_{i=1}^{N_{CELLS}} I_{leak} = \sum_{i=1}^{N_{ROW}(N_{COL}+N_{RC})} I_{leak} \tag{10.8}$$

where I_{leak} is the random variable representing the leakage of a cell, N_{CELLS} is the total number of cells in the critical path, N_{ROW} is the number of rows, and N_{COL} and N_{RC} are the number of columns and redundant columns, respectively. Applying the Central Limit Theorem [129], the distribution of the total leakage can be approximated as a Gaussian with a mean (μ_L) and standard deviation (σ_L) given by

$$\mu_L = N_{\text{CELLS}} \mu_{\text{LCELL}} \quad \text{and} \quad \sigma_L^2 = N_{\text{CELLS}} \sigma_{\text{LCELLS}}^2 \tag{10.9}$$

To quantify the effect of the leakage distribution on the statistical design of the SRAM array, we have defined the probability (P_L) that the total memory leakage will meet a given bound as

$$P_{LeakMem} = P(I_{LeakMem} \leq I_{LMAX}) = \Phi(\frac{I_{LMAX} - \mu_{LMEM}}{\sigma_{LMEM}}) \quad (10.10)$$

10.4.4 Impact of Transistor Threshold Voltage (V_{th}) and Temperature (T) on Leakage Power

Let P_{leak} and I_{leak} be the chip leakage power and current under V_{th} variation, and P^0_{leak} and I^0_{leak} be the same parameters when there is no variation. The expected value of the ratio of post-variation and pre-variation leakage is [169]:

$$P_{leak/}P^0_{leak} = I_{leak}/I^0_{leak} = e^{(q\sigma/\eta kT)^2/2} \quad (10.11)$$

which implies that the increase in the chip's leakage power and current due to V_{th} variation depends on the standard deviation σ of V_{th}. Figure 10.17 plots the relative power as a function of σ. It increases rapidly as σ goes up.

Another important factor affecting leakage power is temperature. Temperature effects are important because leakage current depends exponentially on temperature, and future operating temperatures may exceed 100°C [1]. Figure 10.18 shows how the relative leakage power changes as a function of temperature, for different threshold voltages at 125 °C. Leakage power increases dramatically with temperature (∼4X from 27 °C to 125 °C). In addition, we observe

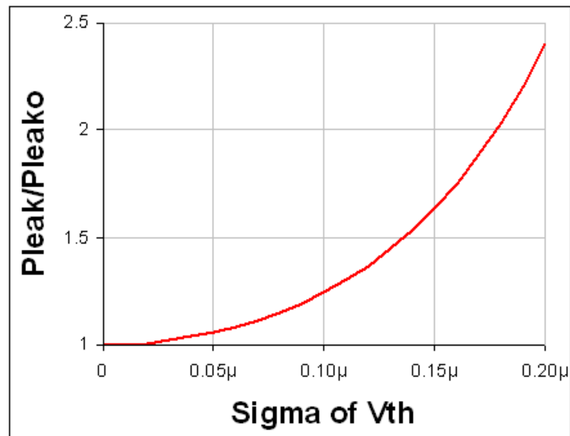


Figure 10.17: Relative leakage power in the 16-nm SRAM chip as a function of V_{th}'s σ. V_{th0} is 0.220V at 125°C.

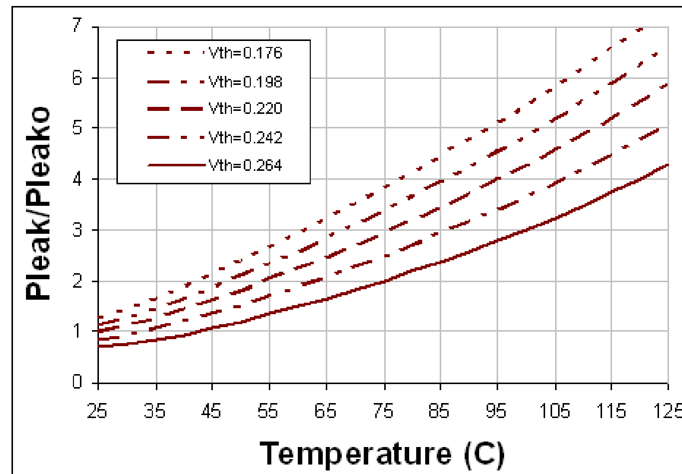


Figure 10.18: Relative leakage power versus temperature for different threshold voltages at 125°C. We use V_{th0}=0.220V at 125°C for our 16-nm design.

that the leakage dependence on the threshold voltage is significant. For different V_{th} values (different lines in Figure 10.18), the leakage changes significantly.

Considering the combined effect of temperature on leakage current and power (shown in Figures 10.18, 10.17, 10.15(c), and 10.15(d)) and on frequency (shown in Figure 10.13), it is reasonable to anticipate that temperature will have a small impact on delay, a dramatic impact

on leakage current, and a considerable impact on the performance and yield of 16-nm SRAM.

10.4.5 Simulation Results for Power, Leakage, and Energy

Figure 10.19 illustrates the leakage power and dynamic power for different SRAM sizes. It also shows the access-time trace to facilitate the comparison between the power consumption and the delay. From the traces on this plot, it is reasonable to deduce that the static and dynamic power increase at almost the same rate, even if their magnitudes are substantially different (the magnitude of the static power is considerably lower than the magnitude of the dynamic power for any given SRAM size). This desirable result is mostly due to the use of architectural techniques which switch the vast majority of the SRAM circuits from active mode to standby mode, thus minimizing the leakage current.

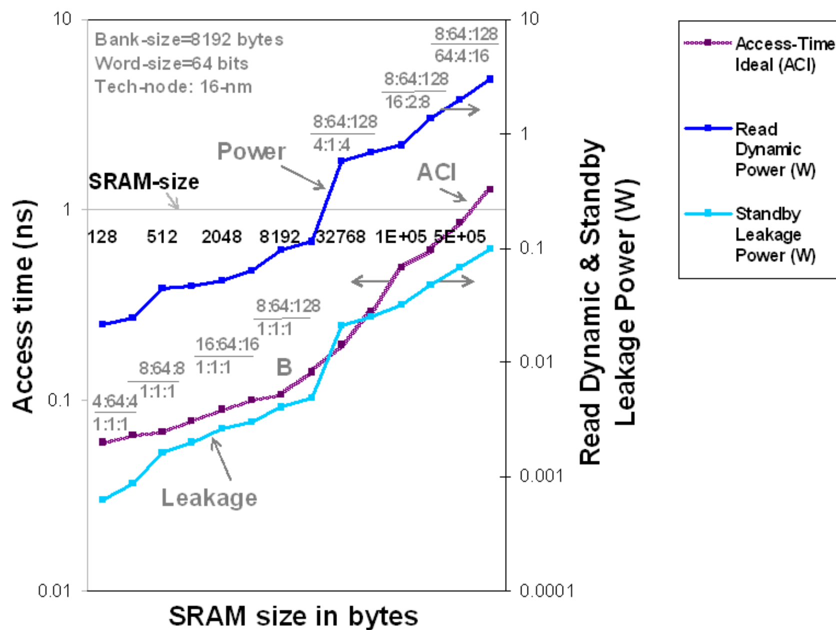


Figure 10.19: Read Dynamic Power, Standby Leakage Power, and Ideal Access-time (ACI) for different SRAM sizes in our 16-nm design.

Figure 10.20 shows the leakage power, dynamic power, and total power, as well as the energy consumption, for different SRAM sizes. It is interesting to see that the magnitude of the leakage current, and therefore the variation, is obscured by the dynamic power, as the heavy red square dots confirm. The energy trace shows a higher rate of increase as compared to the other traces in Figure 10.20—which cautions designers against the use of battery-operated larger-sized SRAM circuits.

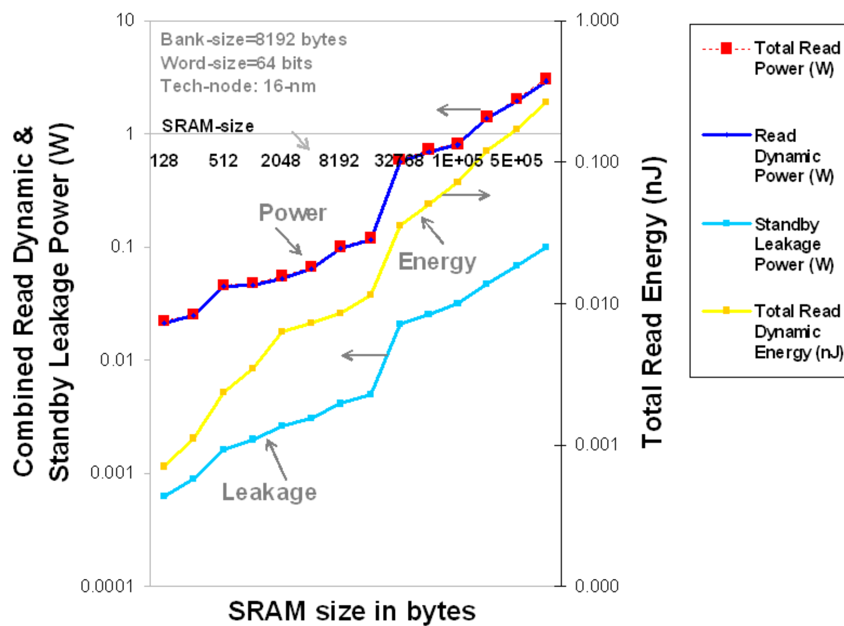


Figure 10.20: Illustrating the combined "Read Dynamic Power + Standby Leakage Power" (shown in red squares, composed of Read Dynamic Power and Standby Leakage Power, shown separately) and the Total Read Dynamic Energy for different SRAM sizes in our 16-nm design.

Figure 10.21 plots the total energy and the access-time for different SRAM sizes to facilitate the comparison between the energy consumption and the delay.

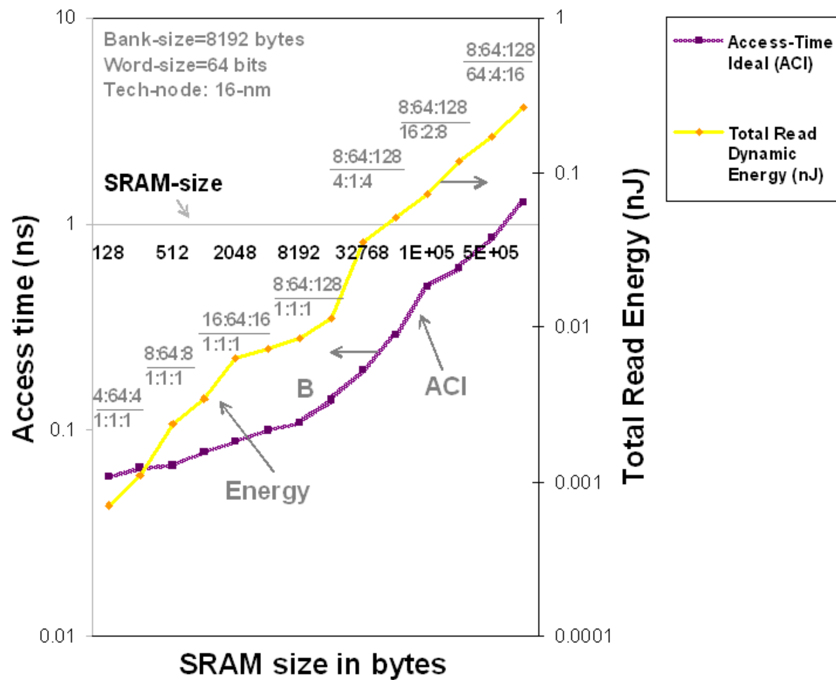


Figure 10.21: Total Read Dynamic Energy and Ideal Access-time (ACI) for different SRAM sizes in our 16-nm design.

10.4.6 Probability Distribution of Total Power

As mentioned in Section 10.4.1 (Power Overview), each of the components of the total power (P_{dyn}, P_{dp}, and P_{stat}) is influenced by process and operation parameters (V_{th}, L, and V_{dd}) and their variation. For example, the dynamic power variation is influenced quadratically by V_{dd} and linearly by the capacitance C_L—which in turn is influenced by the channel length dimensions (W and L) and by the oxide thickness (t_{ox}). Similarly, the variation of the leakage current component of the static power is impacted quadratically by V_{dd}, V_{th}, and T. Thanks to new advanced techniques (such as throttle, etc. discussed in Chapter 7), the impact of the quadratic/exponential variation of the supply voltage on the power consumption can be minimized. Minimization of the larger exponential impact of V_{th} and T on the leakage current still

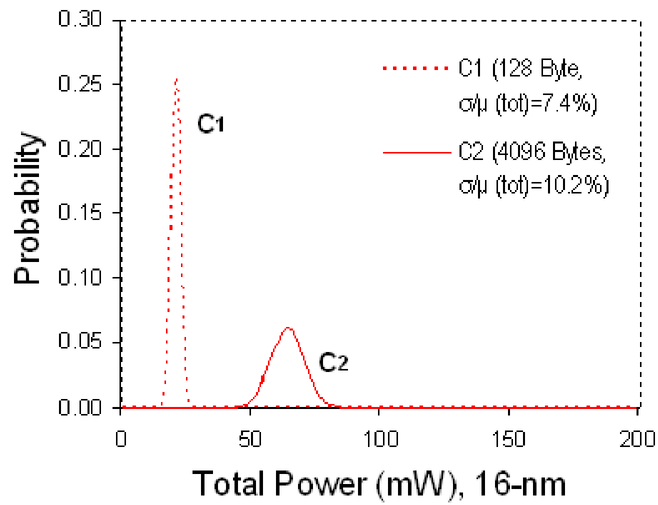
remains one of the challenges for future technology nodes, such as 16-nm. Figure 10.22 shows the probability distribution of the total power for four different SRAM sizes, with the assumed parameter variations presented in Section 9.2 (i.e. independent WID variations of 8.98% for V_{th}, 4.84% for L, and 2% for V_{dd} and independent D2D variations of 4.01% for V_{th} and L, and 2% for V_{dd}). The curves are shown in two charts (a) and (b) to avoid cluttering.

Several observations follow from Figure 10.22

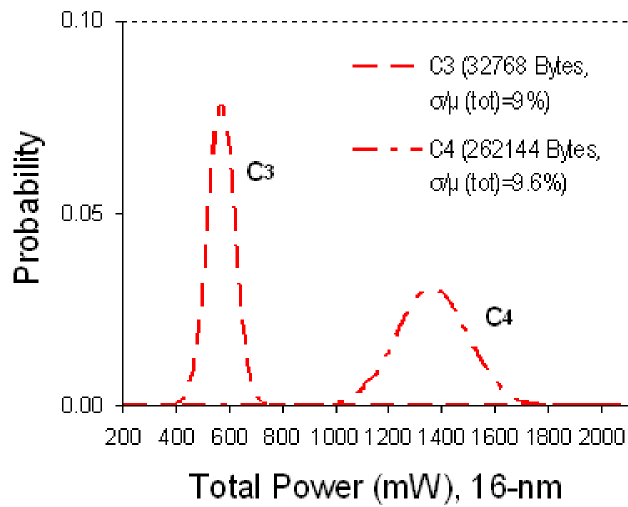
1. SRAM total power and its variation is a function of SRAM size and organization.

2. The probability of power variation is directly related to the length of the row. For example, curve C_1, with a row-width of 4×64 has the lowest deviation, while the curve C_2, with a row-width of 32×64, has the highest deviation.

3. Comparing the different SRAM sizes, it can be deduced that a larger area only slightly increases the probability variation. This holds not just for power, but for delay as well. The plot shows there is only a slight difference between the sigma of an SRAM having 4 banks (4banks×8Kbytes=32768KB) and an SRAM having 32 banks (32banks×8Kbytes= 262144KB). This means the impact of larger area on variation is less pronounced than that of long rows. That is, the probability of the systematic mismatch between the cells on the opposite ends of a long row in each bank is higher than the probability of oxide thickness variation of different banks located on a square or close to square shape. As mentioned before, the variations of the cells located at the end of long rows are mainly due to the cumulative impact of V_{th}, L, and V_{dd} variation, but the variation of banks with each other is mainly due to the timing variation due to non-uniform oxide thickness. Due



(a)



(b)

Figure 10.22: The probability distribution of the total power for four different SRAM sizes. Curve C2 shows the largest variation mainly because its corresponding design has a larger row width value than the other three curves.

to the improvement in process technology which assures more uniform oxide thickness, the variation of banks in large SRAMs is expected to be further reduced.

4. Comparing the PDF traces of the total power of the 16-nm node with the 45-nm and

180-nm nodes (not shown) reveals that the variation of SRAM increases with technology scale-down. While the sigma for a 4kB SRAM with a row-width of 32×64 is only about 3.1% for 180-nm and 6.2% for 45-nm technology, it is about 10.2% for 16-nm node.

5. Finally, it is important to mention that, although the relative variation of leakage power (P_{leak}/P^0_{leak}) is higher than the relative variation of dynamic power (P_{dyn}/P^0_{dyn}), typically by a factor of 10 or so (not shown), the magnitude of leakage power variation is usually obscured by the magnitude of dynamic power variation.

10.5 SRAM yield-estimation model

The D2D and WID variations and hence, failure probability (P_F) of SRAM are directly related to the yield of the memory chip [115]. To estimate the yield, we use Monte Carlo simulations for D2D distributions of V_{th}, L, and V_{dd} (assumed to be Gaussian) in our model. An embedded algorithm takes the result of the Monte Carlo simulation, along with the desired maximum power and area, to determine the optimum yield. The algorithm discards the delays that do not meet the maximum allowable power and area, and selects the smallest delay that meets the criteria. For each parameter D2D value ($V_{\text{th D2D}}$, L_{D2D}, and $V_{\text{dd D2D}}$), we estimate the probability failure ($P_F = 1 - CDF$) considering the WID distribution of ΔV_{th}, ΔL, and ΔV_{dd}. Finally, following Mukhopadhyay, the yield is defined as [115]:

$$Yield = 1 - \left(\frac{\sum_{D2D} P_F \left(V_{\text{th D2D}}, L_{D2D}, V_{\text{dd D2D}} \right)}{N_{D2D}} \right) \tag{10.12}$$

where N_{D2D} is the total number of D2D Monte Carlo simulations (i.e., total number of chips). An increase in the WID variation ($\sigma_{V_{\text{th}}}$, σ_L, σ_{Vdd}) increases the memory-failure probability, thereby reducing the yield. This means that, without proper cell transistor sizing and careful choice of SRAM architecture, yield can suffer significantly. For example, using close to the minimum size width for the pull down transistors of each 6T-cell can increase both the read delay and delay variation. Similarly, increasing the number of cells in a column increases the capacitance and leakage current of the bitlines and also increases the access-time, resulting in an increase in P_F and a decline in yield. Hence, for yield enhancement, the cell configurations and the memory architecture need to be optimized, considering given area and power constraints. In this estimation, we have assumed a standard deviation of 4.01% for D2D distribution of V_{th} and L, and 2% for V_{dd}. Table 10.6 shows the yield results for 16-nm 64KB SRAM. A similar trend holds for all other sizes of 16-nm SRAM—which is about 3% and 5% lower than the trend observed in 45-nm and 180-nm technologies, respectively. To quantify the approximation error empirically, we compare the results obtained from our model with the empirical results obtained from our actual transistor-level SRAM circuits. The approximation error is below 8%.

Table 10.6: SRAM yield before and after optimization.

	Architecture	**Area**	**Total Power**	**Access-Time**	**Yield**
Initial Design (scaled from 50-nm) [115]	$\frac{8:64:256}{1:1:1}$	41 mm^2	0.685 W	0.42 ns	57%
Empirically Optimized Designed SRAM	$\frac{64:64:16}{1:1:1}$	44 mm^2	0.720 W	0.29 ns	93%

Part VII

Conclusion

Chapter 11

Summary

Mismatched Metal Oxide Semiconductor (MOS) transistors impact the performance of today‘s scaled technology more than ever because both device dimensions and available signal swing are much reduced. As the devices have become smaller, it is more crucial to predict, minimize, and prevent process variations, which affect device performance, reliability, stability, and power consumption. Equally important is selecting the optimum organization/architecture in the design of SRAM to minimize the access time, and therefore improve the yield.

As an example, mismatched MOS transistors undermine performance for the 16-nm technology node more than in the 45-nm and 180-nm technologies. Whereas mismatched transistor lengths in the 180-nm SRAM only cause small variations in access time and power, 45-nm SRAM suffers larger variations due to mismatches in transistor *lengths* and *threshold voltages*. These additional variations arise from fluctuations in oxide thickness and also from line-edge roughness (LER). These variations will become even worse in the forthcoming 16-nm case. Therefore, 16-nm SRAM demands more careful transistor sizing, more precise oxide thickness

control, and more advanced lithography techniques, if large access time and power variations are to be avoided. Mismatches in transistors—mainly resulting from oxide thickness, LER, minimum length, and supply voltage—are the most significant variables when predicting access time and leakage power. Unless these variations are analyzed diligently in the early stages of the design process, there is little chance to effectively confront the ever-growing scaling issues.

In this thesis, we presented a literature survey focused on the various circuit techniques that have been proposed to curb process variations and thus improve SRAM access-time and stability while lowering power use. To facilitate a quick review of the most important research on memory modeling, we classified the prior related works into three groups, chronologically: 1. The *Classical Models* (oldest, circa 1990s) that are primarily based on models and equations that take no variability considerations into account. 2. The *more Advanced Models* (coming after the Classical Models) that focus on innovative ways to reduce delay, leakage/dynamic power, or a combination of these two. 3. Finally, the *Current/Recent models* (following the Advanced Models) that are based on the analysis of the effects of variability on memory performance.

We reviewed 6T-cell operation, its design challenges, and the main causes for failure. To combat different variation issues impacting the performance of an on-chip 6T-SRAM of today, and especially of the next generation nodes, we used a two-pronged attack. (1) We introduced a few of our own newly modified circuits (i.e., bitline segmenting, wordline segmenting, etc) and, more importantly, we introduced our proposed model VAR-TX for process and operational variation studies. (2) We employed other models/methodologies, proposed by

some other authors, for temperature and other variation studies. Such a two-pronged approach helped us predict and therefore minimize the impact of all three different types of variations (namely Operational, Fabrication, and Implementation) on the performance of our 6T-SRAMs.

Specifically speaking, we presented a new method for computing the delay distribution of access-time that considers both D2D and architecture-dependent, spatially-correlated WID variations. We proposed a model for D2D and WID device threshold voltage, length, and supply voltage variations and showed how the delay distribution can be efficiently computed using delay sensitivities. Similarly, we presented some recent well-received proposed method that we used to predict the impact of temperature, NBTI, and leakage current on our 16-nm 6T-SRAMs designs.

In addition to presenting the aforementioned effective methods, we briefly covered the most important challenges regarding D2D (inter-die), WID (intra-die), SNM (Static Noise Margin), Major Techniques for Leakage Control in Caches/SRAMs, Soft Error, NBTI (Negative Bias Temperature Instability), HCI (Hot-Carrier Injection), Impact of Temperature on Delay, Power, and Performance, Temperature and Voltage Variation, IR-Drop, EM (electromigration), and L di/dt, Interconnect, Power, Leakage, and Energy Delay, and several others that we believe will significantly impact the future technology nodes beyond 32-nm. While some of these topics were aimed at reducing the variability and increasing stability, reliability, and robustness of 6T-SRAM, some others intended to keep the associated power and energy in check while trying to increase the speed.

In this thesis, we pointed out that the process variations can be classified as systematic or random, where *systematic variation* is deterministic in nature and is caused by the structure

of a particular gate and its topological environment. We find that with increased process scaling, WID variations have become a more dominant portion of the overall variability, meaning that devices on the same die can no longer be treated as identical copies. In our modeling, we showed how to take the impact of systematic D2D and both systematic and random WID variations on circuit performance into consideration and illustrated several plots that compared the impact of D2D and WID variations on access-time and power for 16-nm SRAM with either the 45-nm, 180-nm, or both of these two nodes and demonstrated that the drastic increase in the 1- and 3-sigma of the smaller nodes is mainly due to the increase in the WID variations.

We also presented and analyzed a considerable number of our simulation results regarding access-time, leakage current, and dynamic power to help predict and thereby minimize the impact of process, operation, temperature, and aging variations on SRAM variability. Moreover, with our VAR-TX model, we argued previously published works that suggest that square SRAM always produces minimum delays. We showed that perfectly square banks do not necessarily lead to minimum access-times. Furthermore, and more importantly, we demonstrated how selecting the optimal architecture can increase the yield in SRAM. Finally, by introducing our VAR-TX model we significantly extended and enhanced the older models by adding both an extra dimension of architectural consideration and additional device parameter fluctuation to the analysis, while providing results that are within 8% of Hspice. We tested and validated the accuracy of our proposed approach by comparing our results with Monte Carlo simulation and the access-time method discussed by Mukhopadhyay [115] and VARIUS [169].

In addition to sharing our model and our analytical method for free (through the publication of this thesis), which provides the means to predict the access-time and access-time vari-

ation in current and next-generation on-chip memories, as well as providing a broad overview of the important challenges in SRAM design, we are also making the proposed model software/toolkit VAR-TX freely available, as an extra bonus, through email request to the author: jeffsrad@soe.ucsc.edu. The user can run the VAR-TX program on-line by entering a desired set of four SRAM specifications, provided and explained in more detail by the software. These specifications include SRAM size and shape, number of columns, word-size, and technology node. VAR-TX will return detailed tabulated data that suggests optimized architecture for yield enhancement and an estimated prediction of the associated access-time $T_{ACCESS-TIME}$, and the variation of access-time $\delta_{T_{ACCESS-TIME}}$, all within 30 seconds.

The biggest impact of this thesis is to provide SRAM designers and computer architects with a model (explicitly discussed in Chapter 9 and implicitly discussed throughout this thesis) that is fast and more complete than what is currently available. It is becoming increasingly difficult to optimize SRAM architecture with area constraints while effectively predicting the access time, power consumption, and their associated range of process variations. The semiconductor industry is rife with examples of projects that have been canceled or delayed due to a lack of understanding the complexity involved in the design, verification and simulation. Specifically, our thesis made the following impacts:

★ We proposed a novel hybrid analytical-empirical model VAR-TX that helps predict the minimum delay and/or minimum delay variation in current and next generation on-chip memories.

★ Our VAR-TX model provides a first-order solution to mitigate the effects of increasing

process variations in future technology nodes, while providing results that are within 8% of Hspice.

★ Our VAR-TX model helps predict the optimum architecture that helps maximize the yield.

★ Our VAR-TX model contradicts previously published works that suggest square SRAM always produce minimum delays.

★ Additionally, we presented the access-time and power variations calculated by our model for the future 16-nm node and compared it to those of the recent 45-nm and/or older 180-nm nodes.

★ By publishing this thesis, we are making our proposed modeling methodology freely available to the public. As a bonus, we are also making the associated toolkit/software of our proposed model VAR-TX freely available to the public upon request (email jeff-srad@soe.ucsc.edu). The VAR-TX toolkit predicts the optimum architecture of a 6T-SRAM to achieve maximum speed for given power and area constraints.

★ The proposed model and analysis method that was applied to standard 6T-SRAM in this thesis provides the ground work for its extension to other types of memory such as 8T-, 10T-, or multi-ported SRAM, cache and CAM in a straightforward manner for future work.

★ This thesis gives a broad overview of the important challenges in SRAM design and could be a valuable reference for SRAM designers.

★ By sharing our model and analytical method for free with the VLSI design community,

we are providing a fast and accurate method for long mixed-signal circuit simulations, which will hopefully increase the success of future circuit designs.

Chapter 12

Future Work

Currently, our existing model maximizes the predicted yield while prioritizing speed optimization over power and/or area optimization. The model implements the speed optimization through the examination of different combinations of architectures while minimizing the power and area as much as possible without sacrificing the speed. In future work, the model can be extended to give the user the option of prioritizing the optimization of any of the aforementioned three parameters. Alternatively, the extended version of our model could add two additional user entries, namely, the desired total power and the desired total area, acting as two input constraints.

Furthermore, our proposed model and analysis method was applied to standard 6T-SRAM, although with some modification, extensions to other types of memory such as 8T-, 10T-, or multi-ported SRAM, L1/L2 cache, and CAM are straightforward. For example, based on our modeling and optimization results and according to Wilton‘s findings [189], there is an optimum cache size between the two extremes of direct-mapped cache and set-associative

cache, as defined and discussed in [189, 131], when both miss rate data and process variations are taken into account. This and similar research topics could be investigated in future work, as well.

Our proposed model is not yet portable to other simulators. Some extra work such as adding and/or modifying code has to be made to make VAR-TX portable to other simulators, such as Ultrasim, Wattch, etc.

In this thesis, our transistor modeling and simulations are based on Arizona State University Predictive Technology Models (ASU-PTM) [33]. Future work that might use multi-gate devices in their memory design can use the newly released (June 2012) set of models for multi-gate transistors (PTM-MG), for both high-performance (HP) and low-standby power (LSTP) applications. This new set of transistor models is based on BSIM-CMG, a dedicated model for multi-gate devices. PTM-MG is developed in collaboration with ARM [33].

From among the four types of SRAM failures (Read failure, Write failure, Access failure, and Hold failure), we only considered Access failure in our modeling and analysis of SRAM delay and delay variations in this thesis—simply because Access failure is by far the most influential culprit for chip failure [115]. By such consideration, we were still able to obtain simulation results sufficiently accurate for first order approximation while avoiding prohibitively complex computations. However, it is possible to consider the joint probability failure of all four SRAM failures if higher accuracy at the expense of making computationally expensive models is desired (or required or unavoidable).

In this thesis, among other techniques, we used RC based copper wires, variable-size logical effort buffers (in place of constant size combinational logic), and our modified version

of Rabaey's [141] techniques for bank organization to maximize the speed, minimize the static and dynamic power consumption, and lower the circuit parameter fluctuations. In the future work, such techniques as the ability to model different types of wires, such as RC based wires with different power, delay, and area characteristics and differential low-swing buses can be further examined. Such examination (or research work) can be extended to 3D SRAM with 3D carbon nano-tubes (CNT) router architecture. Similarly, for cache related research works, such techniques as the ability to model Non-Uniform Cache Access (NUCA) in a 3-dimensional (3D) setting for chip multiprocessors that takes into account the effect of network contention during the design space exploration can be investigated with different types of interconnects (i.e., CNT, etc) presented in Section 7.4 of this thesis.

For future work, the use of the tunable replica bits (TRBs) [143] method (discussed in Section 7.2.3) can be repeated for smaller nodes (i.e., 16-nm) by examining the amount of V_{dd} guardband mitigated by this method. For example, measured data on a 16-nm 16KB 8T-array featuring tunable replica bits illustrating a 10% reduction of the operating minimum V_{dd} (Vmin) and the corresponding 8% reduction in array power would indicate the applicability of the TRBs method for smaller nodes as well.

In reviewing some of the recent lithography-related methodologies in Section 2.3, we saw that the move to low-k1 lithography has made it increasingly difficult to print feature sizes that are a small fraction of the wavelength of light. This difficulty has made many of the manufacturing processes still treat a target layout as a fixed requirement for lithography. However, in reality, layout features may vary within certain bounds without violating design constraints. We mentioned that the knowledge of such tolerances, coupled with models for process variability,

can help improve the manufacturability of layout features while still meeting design requirements. Although some authors [15] have already shown that the *shape tolerances* produced by their proposed methodology can be used within a process-window optical proximity correction (PWOPC) flow to reduce delay errors arising from variations in the lithographic process, their work has focused on only one (and not all) of the several manufacturing layers. In future work, shape slack generation (explained in Section 2.3) for other layers such as active area (which defines transistor widths) can also be explored. Since layout tolerances are additive in nature, there is a natural trade-off between the amount of slack that may be apportioned to the various manufacturing layers [15].

Future work could also include the exploration of new interconnect materials and optimization of interconnect dimensions in future technology nodes while meeting such required minimum metrics as bandwidth and energy with constraints on aspect ratio, dielectric constant, maximum crosstalk, yield, etc.

Although there have been several different controversial proposed techniques to alleviate the impact of NBTI induced degradation, from the circuit-level [46, 74, 96, 180, 183] to the architecture-level [11, 38, 78, 155], as we discussed them in Section 6.4, they are mostly at their infancy stage and therefore suggest room for further research work in the future. For example, the analysis of NBTI in greater depth, including the dependence of results on process/technology, power gating, dynamic voltage scaling (DVS), spatial granularities, as well as overheads introduced by the mitigation techniques (i.e. DVS, lifetime awareness, dynamic instruction scheduling, power gating), and parallelizing the numerical simulator to reduce NBTI degradation simulation runtime can be explored.

In closing, it is clear that the need for the skills and expertise that surround silicon implementation, whether at the technology, circuit, system or CAD levels, will continue to be needed and valued as we continue to scale further. The isolation of skills, however, will be increasingly less possible. For example, no longer will it be possible to work on SRAM design without a firm understanding of how the transistors used in the design are created in the lithography/manufacturing process (or stage) and how process and other sources of variations can impact the expected performance of the design. The simple technology abstractions that have worked for many generations like rectangular shapes, Boolean design rules, and constant parameters will not suffice to enable us to push designs to the ultimate levels of performance. Efficient models and/or toolkits (such as VAR-TX) must become more technology aware if they are to take part in solving challenging scaling problems.

Bibliography

[1] International Technology Roadmap for Semiconductor (ITRS), 2011.

[2] Naeemi A. and J. D. Meindl. Conductance modeling of graphene nanoribbons GNR interconnects. *IEEE Electron Device Lett.*, 28(5):428–431, May 2007.

[3] Jaume Abella, Xavier Vera, and Antonio Gonzalez. Penelope: The NBTI-Aware Processor. In *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 40, pages 85–96, Washington, DC, USA, 2007. IEEE Computer Society.

[4] Aseem Agarwal, David Blaauw, Vladimir Zolotov, Savithri Sundareswaran, Min Zhao, Kaushik Gala, and Rajendran Panda. Path-based Statistical Timing Analysis Considering Inter and Intra-die Correlations. In *In Proceedings of ACM/IEEE International Workshop on Timing Issues (TAU)*. ACM/IEEE, 2002.

[5] Kanak Agarwal and Sani Nassif. Statistical analysis of SRAM cell stability. In *Proceedings of the 43rd annual Design Automation Conference*, DAC ‘06, pages 57–62, New York, NY, USA, 2006. ACM.

[6] Kanak Agarwal and Sani Nassif. Statistical analysis of SRAM cell stability. In *Proceedings of the 43rd annual Design Automation Conference*, DAC ‘06, pages 57–62, New York, NY, USA, 2006. ACM.

[7] M.A. Alam. A critical examination of the mechanics of dynamic NBTI for PMOSFEts. In *Electron Devices Meeting, 2003. IEDM ‘03 Technical Digest. IEEE International*, pages 14.4.1–14.4.4, 2003.

[8] Muhammad Ashraful Alam and S. Mahapatra. A comprehensive model of pmos nbti degradation. *Microelectronics Reliability*, 45(1):71–81, 2005.

[9] Bharadwaj S. Amrutur and Mark A. Horowitz. Speed and Power Scaling of SRAM‘s. *The International Symposium on Quality Electronic Design (ISQED)*, 32(2):175–185, February 2000.

[10] D. A. Antoniadis, I. J. Djomehri, K. M Jackson, and S. Miller. Well-Tempered, On line.

[11] Asen Asenov, Andrew R. Brown, John H. Davies, Savas Kaya, and Gabriela Slavcheva. Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs. *IEEE Trans. Electron Devices*, 50(9):1837–1852, September 2003.

[12] Navid Azizi, Andreas Moshovos, and Farid N. Najm. Low-leakage asymmetric-cell SRAM. In *Proceedings of the 2002 international symposium on Low Power Electronics and Design*, ISLPED '02, pages 48–51, New York, NY, USA, 2002. ACM.

[13] Arash Azizi Mazreah, Mohammad T. Manzuri Shalmani, Hamid Barati, Ali Barati, and Ali Sarchami. A Low Power SRAM Base on Novel Word-Line Decoding. *World Academy of Science, Engineering and Technology*, 2(1):149–153, 2008.

[14] Kaustav Banerjee, Shukri J. Souri, Pawan Kapur, and Krishna C. Saraswat. 3-D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration. In *Proceedings of the IEEE*, pages 602–633, 2001.

[15] Shayak Banerjee, Kanak B. Agarwal, Chin-Ngai Sze, Sani Nassif, and Michael Orshansky. A methodology for propagating design tolerances to shape tolerances for use in manufacturing. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '10, pages 1273–1278, 3001 Leuven, Belgium, Belgium, 2010. European Design and Automation Association.

[16] Aditya Bansal, Rama N. Singh, Rouwaida N. Kanj, Saibal Mukhopadhyay, Jin-Fuw Lee, Emrah Acar, Amith Singhee, Keunwoo Kim, Ching-Te Chuang, Sani Nassif, Fook-Luen Heng, and Koushik K. Das. Yield estimation of SRAM circuits using "Virtual SRAM Fab". In *Proceedings of the 2009 International Conference on Computer-Aided Design*, ICCAD '09, pages 631–636, New York, NY, USA, 2009. ACM.

[17] Robert C. Baumann. Soft Errors in Advanced Semiconductor Devices—Part I: The Three Radiation sources. *Device and Materials Reliability, IEEE Transactions on*, 1(1):17–22, March 2001.

[18] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula. New Predictive modeling of the NBTI effect for reliable design. In *in Proc. IEEE Custom Integr. Circuits Conf*, pages 189–192, Dept. of Electr. Eng., Arizona State Univ., Tempe, AZ, 2006.

[19] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula. Predictive Modeling of the NBTI Effect for Reliable Design. In *in Proc. IEEE Custom Integr. Circuits Conf*, pages 189–192, 2006.

[20] A. Bhavanagarwala, X. Tang, and J. Meindl. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *IEEE Journal of Solid-State Circuits*, 36(4):658–665, 2001.

[21] A. Bhavnagarwala, X. Tang, and J. D. Meindl. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *IEEE J. Solid-State Circuits*, 36(4):658–665, April 2001.

[22] A.J. Bhavnagarwala, Xinghai Tang, and J.D. Meindl. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *Solid-State Circuits, IEEE Journal of*, 36(4):658–665, April 2001.

[23] Aveek Bid, Achyut Bora, and A. K. Raychaudhuri. Temperature Dependence of the Resistance of Metallic Nanowires of Diameter $\geq$ 15nm: Applicability of Bloch-Gruneisen Theorem, 2006.

[24] Li Bingxi and Chunhong Chen. Design Methodology for Electron-Trap Memory Cells. In *Nanotechnology, 2008. NANO '08. 8th IEEE Conference on*, pages 22–24, Dept. of Electr. & Comput. Eng., Univ. of Windsor, Windsor, ON, 2008.

[25] Zhai Bo, D. Blaauw, D. Sylvester, and S. Hanson. A Sub-200mV 6T SRAM in $0.13\mu m$ CMOS. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, Feb 2007, pages 332–333, Michigan Univ., Ann Arbor, MI, 2007.

[26] Shekhar Borkar. Electronics beyond nano-scale CMOS. In *Proceedings of the 43rd annual Design Automation Conference*, DAC '06, pages 807–808, New York, NY, USA, 2006. ACM.

[27] Shekhar Borkar, Tanay Karnik, Siva Narendra, Jim Tschanz, Ali Keshavarzi, and Vivek De. Parameter variations and impact on circuits and microarchitecture. In *Proceedings of the 40th annual Design Automation Conference*, DAC '03, pages 338–342, New York, NY, USA, 2003. ACM.

[28] Shekhar Borkar, Tanay Karnik, Siva Narendra, Jim Tschanz, Ali Keshavarzi, and Vivek De. Parameter variations and impact on circuits and microarchitecture. In *DAC '03: Proceedings of the 40th conference on Design automation*, pages 338–342, New York, NY, USA, 2003. ACM Press.

[29] K.A. Bowman, S.G. Duvall, and J.D. Meindl. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. *Solid-State Circuits, IEEE Journal of*, 37(2):183 – 190, February 2002.

[30] B. H. Calhoun and A. P. Chandrakasan. A 256kb Subthreshold SRAM in 65nm CMOS. In *Proc. of International Solid State Circuits Conference*, February 2006, pages 628–629, 2006.

[31] Andrea Calimera, Enrico Macii, and Massimo Poncino. Nbti-aware power gating for concurrent leakage and aging optimization. In *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, ISLPED '09, pages 127–132, New York, NY, USA, 2009. ACM.

[32] Andrea Calimera, Enrico Macii, Massimo Poncino, and R.Ĩris Bahar. Temperature-insensitive synthesis using multi-vt libraries. In *Proceedings of the 18th ACM Great*

Lakes symposium on VLSI, GLSVLSI '08, pages 5–10, New York, NY, USA, 2008. ACM.

[33] Yu Cao and et al. ASU Predictive Technology Model (PTM), 2011.

[34] Tuck-Boon Chan, John Sartori, Puneet Gupta, and Rakesh Kumar. On the efficacy of NBTI mitigation techniques. In *DATE*, pages 932–937, 2011.

[35] I. Chang, J-J. Kim, S. Park, and K. Roy. A 32kb 10T Subthreshold SRAM Array with Bit-Interleaving and Differential Read Scheme in 90nm CMOS. *Proc. of International Solid State Circuits Conference*, 44(2):650–658, February 2009.

[36] L. Chang, D.M. Fried, J. Hergenrother, J.W. Sleight, R.H. Dennard, R.K. Montoye, L. Sekaric, S.J. McNab, A.W. Topol, C.D. Adams, K.W. Guarini, and W. Haensch. Stable SRAM cell design for the 32 nm node and beyond. In *VLSI Technology, 2005. Digest of Technical Papers. 2005 Symposium on*, June 2005, pages 128–129, IBM Semicond. R&D Center, IBM Semicond. R&D Center, Hopewell Junction, NY, USA, 2005.

[37] Ryu Changsup, Kee-Won Kwon, A.L.S. Loke, Lee Haebum, T. Nogami, V.M. Dubin, R.A. Kavari, G.W. Ray, and S.S. Wong. Microstructure and reliability of copper interconnects. *Electron Devices, IEEE Transactions on*, 46(6):1113–1120, January 1999.

[38] Robert Chau, Suman Datta, Mark Doczy, Jack Kavalieros, and Matthew Metz. Gate dielectric scaling for high-performance CMOS: from SiO2 to High-K. In *In IWGI, 2003. 84*, 2004.

[39] Xiaoming Chen, Yu Wang, Yu Cao, Yuchun Ma, and Huazhong Yang. Variation-aware supply voltage assignment for minimizing circuit degradation and leakage. In *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, ISLPED '09, pages 39–44, New York, NY, USA, 2009. ACM.

[40] B. Cheng, S. Roy, and A. Asenov. The impact of random doping effects on CMOS SRAM cell. In *European Solid State Circuits Conf*, pages 219–222, 2004.

[41] H. Cho, K. H. Koo, P Kapur, and K. C. Saraswat. The delay, energy, and bandwidth comparisons between copper, carbon nanotube, and optical interconnects for local and global wiring application. In *Intl. Interconnect Technology Conference*, page 135, 2007.

[42] G. F. Close, S. Yasuda, B. Paul, S. Fujita, and H.-S. P. Wong. 1-GHz integrated circuit with carbon nanotube interconnects and silicon transistors. *Nano Letters*, 2:706–709, February 2008.

[43] Nicolas B. Cobb, Avideh Zakhor, and Eugene Miloslavsky. Mathematical and CAD framework for proximity correction. *SPIE*, 2726:208–222, June 1996.

[44] J. A. Croon, A. R. Brown, A. Asenov, D. Magot, and T. Linton. Line edge roughness: Characterization, modeling and impact on device behavior. In *Proc. IEDM*, pages 307–310, 2002.

[45] W.J. Dally and J.W. Poulton. *Digital Systems Engineering*. Cambridge University Press, 2008.

[46] Rajat Chaudhry David Blaauw, Sanjay Pant and Rajendran Panda. *Design and Analysis of Power Supply Networks*. CRC Press, 2005.

[47] J. et al. Davis. A 5.6 GHz 64 KB dual-read data cache for the POWER6 processor. In *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pages 2564–2571. IEEE International, September 2006.

[48] Solido Design. Solido Design Releases Worldwide Variation-Aware Custom IC Design Survey Results.

[49] Artem Dinaburg. Bitsquatting: DNS Hijacking without Exploitation, 2011.

[50] O. I. et. al. Dosunmu. High-Speed Resonant Cavity Enhanced Ge Photodetectors on Reflecting Si Substrates for 1550nm Operation. *Technology Letters*, 17(1):175–177, January 2005.

[51] Richard C.D̃orf (ed). *The Electrical Engineering Handbook*. CRC Press, 1st edition, 1993.

[52] P Falferi, R Mezzena, M Mck, and A Vinante. Cooling fins to limit the hot-electron effect in dc SQUIDs. *Journal of Physics: Conference Series 97*, 2008.

[53] Krisztián Flautner, Nam Sung Kim, Steve Martin, David Blaauw, and Trevor Mudge. Drowsy caches: simple techniques for reducing leakage power. *SIGARCH Comput. Archit. News*, 30(2):148–157, May 2002.

[54] Bradley Geden. Understand and Avoid Electromigration (EM) & IR-drop in Custom IP Blocks, 2011.

[55] M. Geetha Priya, K. Baskaran, and D. Krishnaveni. A Novel Leakage Power Reduction Technique for CMOS VLSI Circuits. *European Journal of Scientific Research*, 74(1):96–105, 2012.

[56] A K Geim and K S Novoselov. The rise of graphene. *Nature Mater.*, 6(cond-mat/0702595):183–191, Feb 2007.

[57] Mentor Graphics. *Calibre nmOPC Manual*, 2009.

[58] T. Grasser, B. Kaczer, P. Hehenberger, W. Gos, R. O. Connor, H. Reisinger, W. Gustin, and C. Schlunder. Simultaneous extraction of recoverable and permannent components contributing to bias-temperture instability. In *in Proc. Int. Electron Devices Meeting*, pages 801–804, TU Wien, Vienna, 2007.

[59] P. Gupta, A. Kahng, Y. Kim, S. Shah, and D. Sylvester. Modeling of non-uniform device geometries for post-lithography circuit analysis. *SPIE*, 6156:61560U.1–61560U.10, June 2006.

[60] P. Gupta, A. B. Kahng, P. Sharma, and D. Sylvester. Gate-length biasing for runtime-leakage control. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 25(8):1475–1485, November 2006.

[61] Puneet Gupta, Andrew B. Kahng, Youngmin Kim, and Dennis Sylvester. Self-Compensating Design for Reduction of Timing and Leakage Sensitivity to Systematic Pattern-Dependent Variation. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 26(9):1614–1624, 2007.

[62] Chris Halford. IR-Drop Analysis, 2009.

[63] R. Heald and P. Wang. Variability in sub-100nm SRAM designs. *Computer-Aided Design, International Conference on*, 0:347–352, 2004.

[64] R. Ho, T. Ono, R.-D. Hopkins, A. Chow, J. Schauer, F. Y. Liu, and R. Drost. High speed and low energy capacitively driven on-chip wires. *J. Solid State Circuits*, 43(1):52–60, January 2008.

[65] Cho Hoyeol, Koo Kyung-Hoae, P. Kapur, and K.C. Saraswat. Performance Comparisons Between Cu/Low-K Carbon-Nano-tube, And Optics for Future On-Chip Interconnects. *Electron Device Letters, IEEE*, 29(1):122–124, January 2008.

[66] V. Huard, C. Parthasarathy, N. Rallet, C. Guerin, M. Mammase, D. Barge, and C. Ouvrard. New characterization and modeling approach for NBTI degradation from transistor to product level. In *in Proc. Int. Electron Devices Meeting*, pages 797–800, STMicroelectronics, Crolles, 2007.

[67] Luong Dinh Hung, Masahiro Goshima, and Shuichi Sakai. SEVA: A Soft-Error- and Variation-Aware Cache Architecture. In *PRDC '06*, pages 47–54, 2006.

[68] Y. I. Ismail and E. G. Friedman. Effects of Inductance on the Propagation delay and Repeater Insertion in VLSI Circuits: A Summary. *IEEE Circuits and Systems Magazine*, pages 24–28, September 2003.

[69] Xie Jianyong, Chung Daehyun, M. Swaminathan, M. Mcallister, A. Deutsch, Jiang Lijun, and B.J. Rubin. Effect of system components on electrical and thermal characteristics for power delivery networks in 3D system integration. In *Electrical Performance of Electronic Packaging and Systems, 2009. EPEPS '09. IEEE 18th Conference on*, pages 113–116, Sch. of Electr. & Comput. Eng., Georgia Inst. of Technol., Atlanta, GA, USA, 2009.

[70] Rajiv Joshi, Rouwaida Kanj, Keunwoo Kim, Richard Williams, and Ching-Te Chuang. A floating-body dynamic supply boosting technique for low-voltage sram in nanoscale PD/SOI CMOS technologies. In *Proceedings of the 2007 international symposium on Low power electronics and design*, ISLPED '07, pages 8–13, New York, NY, USA, 2007. ACM.

[71] Rajiv Joshi, Rouwaida Kanj, Sani Nassif, Donald Plass, Yuen Chan, and Ching-Te Chuang. Statistical Exploration of the Dual Supply Voltage Space of a 65nm PD/SOI CMOS SRAM Cell. In *Proceeding of the 36th European Solid-State Device Research Conference (ESSDERC)*, pages 315–318, 2006.

[72] R.V. Joshi, S. Mukhopadhyay, Y.H. Plass, D.W. amd Chan, Chuang Ching-Te, and A. Devgan. Variability analysis of sub-100 nm PD/SOI CMOS SRAM cell. In *Europe Solid State Circuits Conf*, pages 211–214, 2004.

[73] K. Kanda, K. Nose, H. Kawaguchi, and T. Sakurai. Design impact of positive temperature dependence on drain current in sub-1-V CMOS VLSIs. *Solid-State Circuits, IEEE Journal of*, 36(10):1559–1564, October 2001.

[74] Kunhyuk Kang, Sang Phill Park, Kaushik Roy, and Muhammad A. Alam. Estimation of statistical variation in temporal NBTI degradation and its impact on lifetime circuit performance. In *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, ICCAD '07, pages 730–734, Piscataway, NJ, USA, 2007. IEEE Press.

[75] P. Kapur, J.P. McVittie, and K.C. Saraswat. Technology and reliability constrained future copper interconnects-part I: Resistance modeling. *Electron Devices, IEEE Transactions on*, 49(4):590–597, April 2002.

[76] P. Kapur and K.C. Saraswat. Comparisons between electrical and optical interconnects for on-chip signaling. In *Interconnect Technology Conference, 2002. Proceedings of the IEEE 2002 International*, pages 89–91, Dept. of Electr. Eng., Stanford Univ., CA, 2002.

[77] Tanay Karnik, Shekhar Borkar, and Vivek De. Sub-90nm technologies: challenges and opportunities for CAD. In *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, ICCAD '02, pages 203–206, New York, NY, USA, 2002. ACM.

[78] Ulya R. Karpuzcu, Brian Greskamp, and Josep Torrellas. The BubbleWrap many-core: popping cores for sequential acceleration. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, pages 447–458, New York, NY, USA, 2009. ACM.

[79] Stefanos Kaxiras, Zhigang Hu, and Margaret Martonosi. Cache decay: exploiting generational behavior to reduce cache leakage power. *SIGARCH Comput. Archit. News*, 29(2):240–251, May 2001.

[80] John Keane and Chris H. Kim. Transistor Aging, 2011.

[81] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De. Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS ICs. In *Proceedings of the 2001 international symposium on Low power electronics and design*, ISLPED '01, pages 207–212, New York, NY, USA, 2001. ACM.

[82] Ali Keshavarzi, Kaushik Roy, and Charles F. Hawkins. Intrinsic Leakage in Low Power Deep Submicron CMOS ICs. In *Proceedings of the 1997 IEEE International Test Conference*, ITC '97, pages 146–, Washington, DC, USA, 1997. IEEE Computer Society.

[83] O. Kibar, D. A. V. Blerkom, C. Fan, and S. C. Esener. Power Minimization and Technology Comparisons for Digital Free-Space Optoelectronic Interconnects. *IEEE J. of Lightwave Technol.*, 17(4):546–554, April 1999.

[84] Daeyeon Kim, Vikas Chandra, Robert Aitken, David Blaauw, and Dennis Sylvester. An adaptive write word-line pulse width and voltage modulation architecture for bit-interleaved 8T SRAMs. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, ISLPED '12, pages 91–96, New York, NY, USA, 2012. ACM.

[85] Sami Kirolos, Yehia Massoud, and Yehea I. Ismail. Power-supply-variation-aware timing analysis of synchronous systems. In *International Symposium on Circuits and Systems (ISCAS 2008), 18–21 May 2008, Sheraton Seattle Hotel, Seattle, Washington, USA*, pages 2418–2421. IEEE, 2008.

[86] Sachiko Kobayashi, Suigen Kyoh, Toshiya Kotani, Satoshi Tanaka, and Soichi Inoue. Automated hot-spot fixing system applied for metal layers of 65 nm logic devices. In *Proceedings of the SPIE*, page 61560U.1, Toshiba Corp. Semiconductor Co. (Japan), 2006.

[87] Kyung-Hoae Koo. *Comparison study of future on-chip interconnects for high performance VLSI applications*. Re-distributed by Stanford University under license with the author, 2011.

[88] Kyung-Hoae Koo, Hoyeol Cho, P. Kapur, and K.C. Saraswat. Performance Comparisons Between Carbon Nanotubes, Optical, and Cu for Future High-Performance On-Chip Interconnect Applications. *Electron Devices, IEEE Transactions on*, 54(12):3206–3215, dec. 2007.

[89] A. T. Krishnan, C. Chancellor, S. Chakravarthi, P. E. Nicollian, V. Reddy, and A. Varghese. Material dependence of hydrogen diffusion: Implication for NBTI degradation. In *in Proc. IEEE Int. Electron Devices Meeting*, pages 688–691, 2005.

[90] Jente B. Kuang, Jeremy D. Schaub, Fadi H. Gebara, Dieter F. Wendel, Thomas Fröhnel, Sudesh Saroop, Sani R. Nassif, and Kevin J. Nowka. The Design and Characterization of a Half-Volt 32 nm Dual-Read 6T SRAM. *IEEE Trans. on Circuits and Systems*, 58-I(9):2010–2016, 2011.

[91] Jaydeep P. Kulkarni, Keejong Kim, Sang Phill Park, and Kaushik Roy. Process variation tolerant SRAM array for ultra low voltage applications. In *Proceedings of the 45th annual Design Automation Conference*, DAC '08, pages 108–113, New York, NY, USA, 2008. ACM.

[92] Jaydeep P. Kulkarni, Keejong Kim, Sang Phill Park, and Kaushik Roy. Process variation tolerant SRAM array for ultra low voltage applications. In *Proceedings of the 45th annual Design Automation Conference*, DAC '08, pages 108–113, New York, NY, USA, 2008. ACM.

[93] R. Kumar and V. Kursun. Supply and Threshold Voltage Optimization for Temperature Variation Insensitive Circuit Performance: A Comparison. In *SOC Conference, 2006 IEEE International*, pages 89–90, Dept. of Electr. & Comput. Eng., Univ. of Wisconsin -Madison, Madison, WI, 2006.

[94] Rajesh Kumar and Glenn Hinton. A family of 45nm IA processors. In *ISSCC*, pages 58–59. IEEE, 2009.

[95] Sanjay V. Kumar, Chris H. Kim, and Sachin S. Sapatnekar. An analytical model for negative bias temperature instability. In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, ICCAD '06, pages 493–496, New York, NY, USA, 2006. ACM.

[96] Sanjay V. Kumar, Chris H. Kim, and Sachin S. Sapatnekar. Adaptive techniques for overcoming performance degradation due to aging in digital circuits. In *Proceedings of the 2009 Asia and South Pacific Design Automation Conference*, ASP-DAC '09, pages 284–289, Piscataway, NJ, USA, 2009. IEEE Press.

[97] John Lee and Puneet Gupta. Incremental gate sizing for late process changes. In *ICCD*, pages 215–221, 2010.

[98] Xiaoyao Liang, Kerem Turgay, and David Brooks. Architectural power models for SRAM and CAM structures based on hybrid analytical/empirical techniques. In *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, ICCAD '07, pages 824–830, Piscataway, NJ, USA, 2007. IEEE Press.

[99] C.M. Lilley, M. Bode, and R.S. Divan. Electrical Properties of Cu Nanowires. In *Nanotechnology, 2008. NANO '08. 8th IEEE Conference on*, pages 549–552, Dept. of Mech. & Ind. Eng., Univ. of Illinois at Chicago, Chicago, IL, 2008.

[100] J. Lohstroh, E. Seevinck, and J. Groot. Worst-case static noise margin criteria for logic circuits and their mathematical equivalence. *IEEE Journal of Solid-State Circuits*, 18(6):803–806, 1983.

[101] William Lowrie. Electrical resistivity and conductivity, 2012.

[102] C. Mack. *Fundamental Principles of Optical Lithography: The Science of Microfabrication*. Wiley Publishers, 2008.

[103] N. Maeda, Y. Shinozaki, Y. Shimazaki, K. Nii, S. Shimada, K. Yanagisawa, and T. Kawahara. 90-nm process-variation adaptive embedded SRAM modules with power-line-floating write technique. *IEEE Journal of Solid-State Circuits*, 41:705–711, 2006.

[104] S. Mahapatra, P. B. Kumar, and M. A. Alam. Investigation and modeling of interface and bulk trap generation during negative bias temperature instability of p-MOSFETS. *IEEE Trans. Electron Devices*, 51(9):1371–1379, September 2004.

[105] S. Mahapatra, D. Saha, D. Varghese, and P. B. Kumar. On the generation and recovery of interface traps in mosfets subjected to NBTI, FN, and HCI stress. *IEEE Trans. Electron Devices*, 53(6):1583–1592, 2006.

[106] J. Maiz. Characterization of multi-bit soft error events in advanced SRAMs. In *Electron Devices Meeting, 2003. IEDM '03 Technical Digest. IEEE International*, pages 21.4.1–21.4.4, Intel Corp., Hillsboro, OR, USA, 2003.

[107] Vikas Mehrotra, Shiou Lin Sam, Duane Boning, Anantha Chandrakasan, Anantha Ch, Rakesh Vallishayee, and Sani Nassif. A Methodology for Modeling the Effects of Systematic Within-Die Interconnect and Device Variation on Circuit Performance. In *In Proc. 37 th Design Automation Conference*, pages 172–175, 2000.

[108] Eisse Mensink, Daniël Schinkel, Eric Klumperink, Ed Tuijl van, and Bram Nauta. A 0.28pJ/b 2Gb/s/ch Transceiver in 90nm CMOS for 10mm On-Chip interconnects. In *IEEE International Solid-State Circuits Conference, ISSCC 2007: Digest of Technical Papers*, page 414, February 2007.

[109] D.A.B Miller. Rationale and challenges for optical interconnects to electronic chips. *Proceedings of the IEEE*, 88(6):728–749, June 2000.

[110] Gu Ming, Yang Jun, and Xue Jun. Low power SRAM design using charge sharing technique. In *6th International Conference On Aplication Specific Integrated Circuits (ASIC)*, pages 19–23, 2005.

[111] Baker Mohammad, Martin Saint-Laurent, Paul Bassett, and Jacob Abraham. Cache Design for Low Power and High Yield. In *Proceedings of the 9th international symposium on Quality Electronic Design*, ISQED '08, pages 103–107, Washington, DC, USA, 2008. IEEE Computer Society.

[112] Arkadiy Morgenshtein. Short-Circuit Power Reduction by Using High-Threshold Transistors. *Journal of Low Power Electronics and Applications*, 2(1):69–78, 2012.

[113] S. Mukhopadhyay, H. Mahmoodi, and K. Roy. Modeling and estimation of failure probability due to parameter variation in nano-scale SRAMs for yield enhancement. In *in Dig. Tech. Papers VLSI Circuit Symp*, pages 64–67, Honolulu, HI, 2004.

[114] S. Mukhopadhyay, H. Mahmoodi, and K. Roy. Statistical design and optimization of SRAM cell for yield enhancement. In *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, ICCAD '04, pages 10–13, Washington, DC, USA, 2004. IEEE Computer Society.

[115] S. Mukhopadhyay, H. Mahmoodi, and K. Roy. Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 24(12):1859–1880, November 2006.

[116] Saibal Mukhopadhyay, Arijit Raychowdhury, and Kaushik Roy. Accurate estimation of total leakage current in scaled CMOS logic circuits based on compact current modeling. In *Proceedings of the 40th annual Design Automation Conference*, DAC '03, pages 169–174, New York, NY, USA, 2003. ACM.

[117] A. Naeemi and J. D. Meindl. Design and Performance Modeling for Single-Wall Carbon Nanotubes as Local, Semi-global and Global Interconnects in Gigascale Integrated Systems. *Trans. on Electron Device*, 54:26–37, January 2007.

[118] A. Naeemi, R. Sarvari, and J.D. Meindl. Performance comparison between carbon nanotube and copper interconnects for GSI. In *Electron Devices Meeting, 2004. IEDM Technical Digest. IEEE International*, pages 699–702, Georgia Inst. of Technol., Atlanta, GA, USA, 2004.

[119] Y. Nakagome, M. Horiguchi, T. Kawahara, and K. Itoh. Review and future prospects of low-voltage RAM circuits. *IBM J. Res. Dev.*, 47(5–6):525–552, September 2003.

[120] S. Narendra, M. Haycock, V. Govindarajulu, V. Erraguntla, H. Wilson, S. Vangal, A. Pangal, E. Seligman, R. Nair, A. Keshavarzi, B. Bloechel, G. Dermer, R. Mooney, N. Borkar, S. Borkar, and V. De. 1.1 V 1 GHz communications router with on-chip body bias in 150 nm CMOS. In *Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International*, pages 270–466, 2002.

[121] R. Nassif. Modeling and analysis of manufacturing variations. In *in Proc. IEEE Conference on Custom Integrated Circuits*, pages 223–228, San Diego, CA, 2001.

[122] Verma Naveen and A.P. Chandrakasan. A 65nm 8T Sub-Vt SRAM Employing Sense-Amplifier Redundancy. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, Feb. 2007, pages 328–329, Massachusetts Inst. of Technol., Cambridge, MA, 2007.

[123] Koji Nii, Hiroshi Makino, Yoshiki Tujihashi, Chikayoshi Morishima, Yasushi Hayakawa, Hiroyuki Nunogami, Takahiko Arakawa, and Hisanori Hamano. A low power SRAM using auto-backgate-controlled MT-CMOS. In *Proceedings of the 1998 international symposium on Low power electronics and design*, ISLPED '98, pages 293–298, New York, NY, USA, 1998. ACM.

[124] U.S.D̃ept. of Defense. *Integrated circuits (microcircuits) manufacturing, general specification*. Std, 2007.

[125] A. K. Okyay, A. M. Nayfeh, A. Marshall, T. Yonehara, P. C. McIntyre, and K.C. Saraswat. Ge on Si by Novel Heteroepitaxy for High Efficiency Near Infra-red Photodetection. In *Conference on Lasers and Electro-Optics (CLEO)*, 2006.

[126] M. Orshansky, L. Milor, Chen Pinhong, K. Keutzer, and Hu Chenming. Impact of systematic spatial intra-chip gate length variability on performance of high-speed digital circuits. In *Computer Aided Design, 2000. ICCAD-2000. IEEE/ACM International Conference on*, pages 62–67, California Univ., Berkeley, CA, 2000.

[127] Michael Orshansky, Linda Milor, Pinhong Chen, Kurt Keutzer, and Chenming Hu. Impact of systematic spatial intra-chip gate length variability on performance of high-speed digital circuits. In *Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design*, ICCAD '00, pages 62–67, Piscataway, NJ, USA, 2000. IEEE Press.

[128] Michael Orshansky, Linda Milor, Pinhong Chen, Student Member, Student Member, Kurt Keutzer, and Chenming Hu. Impact of Spatial Intrachip Gate Length Variability on the Performance of High-Speed Digital Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21:544–553, 2002.

[129] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. New York: MacGraw-Hill, New York: MacGraw-Hill:adr, 2002.

[130] Dongkook Park, Soumya Eachempati, Reetuparna Das, Asit K. Mishra, Yuan Xie, Narayanan Vijaykrishnan, and Chita R. Das. MIRA: A Multi-layered On-Chip Interconnect Router Architecture. In *ISCA*, pages 251–261. IEEE, 2008.

[131] David A. Patterson and John L. Hennessy. *Computer Organization and Design - The Hardware / Software Interface (Revised 4th Edition)*. The Morgan Kaufmann Series in Computer Architecture and Design. Academic Press, 2012.

[132] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy. Impact of NBTI on the temporal performance degradation of digital circuits. *IEEE Electron Device Letters*, 26(8):560–562, August 2005.

[133] Bipul C. Paul, Kunhyuk Kang, Haldun Kufluoglu, Muhammad Ashraful Alam, and Kaushik Roy. Temporal performance degradation under NBTI: estimation and design for improved reliability of nanoscale circuits. In *Proceedings of the conference on Design, automation and test in Europe: Proceedings*, DATE '06, pages 780–785, 3001 Leuven, Belgium, Belgium, 2006. European Design and Automation Association.

[134] Marcel J. M. Pelgrom, Aad C. J. Duinmaijer, and Andanton P. G. Welbers. Matching Properties of MOS transistors. *IEEE J. Solid-State Circuits*, 24:1433–1440, 1989.

[135] Jürgen Pille, Dieter F. Wendel, Otto Wagner, Rolf Sautter, Wolfgang Penth, Thomas Fröhnel, Stefan Büttner, Otto A. Torreiter, Martin Eckert, Jose Paredes, David Hrusecky, David Ray, and Miles Canada. A 32kB 2R/1W L1 data cache in 45nm SOI technology for the POWER7TM processor. In *ISSCC*, pages 344–345, 2010.

[136] W. J. Poppe, L. Capodieci, J. Wu, and A. Neureuther. From poly line to transistor: Building BSIM models for non-rectangular transistors. *SPIE*, 6156:235–243, 2006.

[137] Michael Powell, Se-Hyun Yang, Babak Falsafi, Kaushik Roy, and T. N. Vijaykumar. Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories. In *Proceedings of the 2000 international symposium on Low power electronics and design*, ISLPED '00, pages 90–95, New York, NY, USA, 2000. ACM.

[138] J. Puchner and L. Hinh. NBTI reliability analysis for a 90 nm CMOS technology. In *Solid-State Device Research conference, 2004. ESSDERC 2004. Proceeding of the 34th European*, pages 257–260, Technol. R&D, Cypress Semicond., San Jose, CA, USA, 2004.

[139] Kiran Puttaswamy and Gabriel H. Loh. Thermal Herding: Microarchitecture Techniques for Controlling Hotspots in High-Performance 3D-Integrated Processors. In *Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture*, HPCA '07, pages 193–204, Washington, DC, USA, 2007. IEEE Computer Society.

[140] Huifang Qin, Yu Cao, Dejan Markovic, Andrei Vladimirescu, and Jan Rabaey. SRAM leakage suppression by minimizing standby supply voltage. In *in Proc. Int. Symposium on Quality Electronic Design*, pages 55–60, 2004.

[141] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. *Digital Integrated Circuits*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2008.

[142] T. Rahal-Arabi, G. Taylor, M. Ma, and C. Webb. Design and validation of the Pentium III and Pentium 4 processors power delivery. In *VLSI Circuits Digest of Technical Papers, 2002. Symposium on*, pages 220–223, 2002.

[143] Arijit Raychowdhury, Bibiche M. Geuskens, Keith A. Bowman, James W. Tschanz, Shih-Lien Lu, Tanay Karnik, Muhammad M. Khellah, and Vivek K. De. Tunable Replica Bits for Dynamic Variation Tolerance in 8T SRAM Arrays. *J. Solid-State Circuits*, 46(4):797–805, 2011.

[144] Arijit Raychowdhury, Saibal Mukhopadhyay, and Kaushik Roy. A Feasibility Study of Subthreshold SRAM Across Technology Generations. In *Proceedings of the 2005 International Conference on Computer Design*, ICCD '05, pages 417–424, Washington, DC, USA, 2005. IEEE Computer Society.

[145] Sherief Reda and Sani R. Nassif. Analyzing the impact of process variations on parametric measurements: novel models and applications. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '09, pages 375–380, 3001 Leuven, Belgium, Belgium, 2009. European Design and Automation Association.

[146] Ho Ron, T. Ono, R.D. Hopkins, A. Chow, J. Schauer, F.Y. Liu, and R. Drost. High Speed and Low Energy Capacitively Driven On-Chip Wires. *Solid-State Circuits, IEEE Journal of*, 43(1):52–60, January 2008.

[147] Jeren Samandari-Rad, Matthew R. Guthaus, and Richard Hughey. VAR-TX: A variability-aware SRAM model for predicting the optimum architecture to achieve minimum access-time for yield enhancement in nano-scaled CMOS. In *ISQED*, pages 506–515, 2012.

[148] K.C. Saraswat. Scaling of Interconnections, 2003.

[149] Tan S.C. and X.W. Sun. Low power CMOS level shifters by bootstrapping technique. *IEEE Electronics Letters*, 38(16), August 2002.

[150] Dieter K. Schroder. Negative bias temperature instability: What do we understand? *Microelectronics Reliability*, 47(6):841–852, 2007.

[151] Dieter K. Schroder and Jeff A. Babcock. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of Applied Physics*, 94(1):1–18, July 2003.

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

[153] George Sery, Shekhar Borkar, and Vivek De. Life is CMOS: why chase the life after? In *Proceedings of the 39th annual Design Automation Conference*, DAC '02, pages 78–83, New York, NY, USA, 2002. ACM.

[154] Tanmay Shah. FabMem: A Multiported RAM and CAM Compiler for Superscalar Design Space Exploration, 2010.

[155] Yang Shao, Zhen Peng, and Jin-Fa Lee. Thermal-aware DC IR-drop co-analysis using non-conformal domain decomposition methods. *Proc. R. Soc. A 2012 468*, pages 1652–1675, February 2012.

[156] Mohammad Sharifkhani and Manoj Sachdev. Segmented virtual ground architecture for low-power embedded SRAM. *IEEE Trans. Very Large Scale Integr. Syst.*, 15(2):196–205, February 2007.

[157] Sean X. Shi, Peng Yu, and David Z. Pan. A unified non-rectangular device and circuit simulation model for timing and power. In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, ICCAD '06, pages 423–428, New York, NY, USA, 2006. ACM.

[158] Taniya Siddiqua and Sudhanva Gurumurthi. A multi-level approach to reduce the impact of NBTI on processor functional units. In *Proceedings of the 20th symposium on Great lakes symposium on VLSI*, GLSVLSI '10, pages 67–72, New York, NY, USA, 2010. ACM.

[159] Ritu Singhal, Asha Balijepalli, Anupama Subramaniam, Frank Liu, Sani Nassif, and Yu Cao. Modeling and analysis of non-rectangular gate for post-lithography circuit simulation. In *Proceedings of the 44th annual Design Automation Conference*, DAC '07, pages 823–828, New York, NY, USA, 2007. ACM.

[160] Jayanth Srinivasan, Sarita V. Adve, Pradip Bose, Sarita V. Adve Pradip Bose, and Jude A. Rivers. The Case for Lifetime Reliability-Aware Microprocessors. In *In Proc. of the 31st International Symposium on Computer Architecture*, pages 276–287, 2004.

[161] Jayanth Srinivasan, Sarita V. Adve, Pradip Bose, and Jude A. Rivers. Lifetime Reliability: Toward an Architectural Solution. *IEEE Micro*, 25(3):70–80, May 2005.

[162] Richard E. Stallcup, Zachary Cross, William James, and Phuc Ngo. Measuring Static Noise Margin of 65nm Node SRAMS using a 7-Positioner SEM Nanoprobing Technique, 2007.

[163] W. Steinhogl, G. Schindler, and M. Engelhardt. Size-dependent resistivity of metallic wires in the mesoscopic range. *Phys. Review B*, 66(1), January 2002.

[164] Anupama R. Subramaniam, Ritu Singhal, Chi-Chao Wang, and Yu Cao. Design rule optimization of regular layout for leakage reduction in nanoscale design. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, ASP-DAC '08, pages 474–479, Los Alamitos, CA, USA, 2008. IEEE Computer Society Press.

[165] Anupama R. Subramaniam, Ritu Singhal, Chi-Chao Wang, and Yu Cao. Leakage reduction through optimization of regular layout parameters. *Microelectronics Journal*, 43(1):25–33, 2012.

[166] T. Suzuki, H. Yamauchi, Y. Yamagami, K. Satomi, and H. Akamatsu. A Stable 2-Port SRAM Cell Design Against Simultaneously Read/Write-Disturbed Accesses. *Solid-State Circuits, IEEE Journal of*, 43(9):2109–2119, September 2008.

[167] Wada T. Rajan S. and Przybylski S.A. An analytical access time model for on-chip cache memories. *IEEE Journal of Solid-State Circuits*, 27:1147–1156, 1992.

[168] Kim Tae-Hyoung, J. Liu, J. Keane, and C.H. Kim. A High-Density Subthreshold SRAM with Data-Independent Bitline Leakage and Virtual Ground Replica Scheme. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, Feb. 2007, pages 330–331, Minnesota Univ., Minneapolis, MN, 2007.

[169] R. Teodorescu, B. Greskamp, J. Nakano, S. Sarangi, A. Tiwari, and Torrellas J. VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects. *IEEE Transactions on Semiconductor Manufacturing*, 21(1):3–13, February 2008.

[170] William A. Tisdale, Kenrick J. Williams, Brooke A. Timp, David J. Norris, Eray S. Aydil, and X. Y. Zhu. Hot-Electron Transfer from Semiconductor Nanocrystals, 2010.

[171] Abhishek Tiwari and Josep Torrellas. Facelift: Hiding and slowing down aging in multicores. In *Proceedings of the 41st annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 41, pages 129–140, Washington, DC, USA, 2008. IEEE Computer Society.

[172] James Tschanz, Keith Bowman, Steve Walstra, Marty Agostinelli, Tanay Karnik, and Vivek De. Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance. In *VLSI Circuits, 2009 Symposium on*, pages 112–113, Circuits Research Lab, USA, 2009.

[173] James Tschanz, Keith A. Bowman, Shih-Lien Lu, Paolo A. Aseron, Muhammad M. Khellah, Arijit Raychowdhury, Bibiche M. Geuskens, Carlos Tokunaga, Chris Wilkerson, Tanay Karnik, and Vivek De. A 45 nm resilient and adaptive microprocessor core for dynamic variation tolerance. In *ISSCC*, pages 282–283, 2010.

[174] James W. Tschanz, James T. Kao, Siva G. Narendra, Raj Nair, Dimitri A. Antoniadis, Anantha P. Ch, Senior Member, and Vivek De. Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage. *IEEE Journal Of Solid-State Circuits*, 37:1396–1402, 2002.

[175] J.W. Tschanz, S.G. Narendra, Y. Ye, B.A. Bloechel, S. Borkar, and V. De. Dynamic-sleep transistor and body bias for active leakage power control of microprocessors. *Solid-State Circuits, IEEE Journal of*, 38(11):1838–1845, November 2003.

[176] K.R. Vaddina, P. Liljeberg, and J. Plosila. Thermal analysis of on-chip interconnects in multicore systems. In *NORCHIP, 2009*, pages 1–4, Turku Center for Comput. Sci. (TUCS), Turku, Finland, 2009.

[177] Rakesh Vattikonda, Wenping Wang, and Yu Cao. Modeling and minimization of PMOS NBTI effect for robust nanometer design. In *Proceedings of the 43rd annual Design Automation Conference*, DAC '06, pages 1047–1052, New York, NY, USA, 2006. ACM.

[178] Rakesh Vattikonda, Wenping Wang, and Yu Cao. Modeling and minimization of PMOS NBTI effect for robust nanometer design. In *Proceedings of the 43rd annual Design Automation Conference*, DAC '06, pages 1047–1052, New York, NY, USA, 2006. ACM.

[179] Jiajing Wang, Satyanand Nalam, and Benton H. Calhoun. Analyzing static and dynamic write margin for nanometer SRAMs. In *Proceedings of the 13th international symposium on Low power electronics and design*, ISLPED '08, pages 129–134, New York, NY, USA, 2008. ACM.

[180] Michael C. Wang. Low Power Dual Word Line 6-Transistor SRAMs. *Proceedings of the World Congress on Engineering and Computer Science (WCECS 2009)*, I, October 2009.

[181] Po-Kang Wang, Yin Rong, Hsu Kai Yang, and Xizeng shi. Word line segment select transistor on word line current source side, 02 2007.

[182] W. Wang, V. Reddy, B. Yang, V. Balakrishnan, S. Krishnan, and Y Cao. Statistical prediction of circuit aging under process variations. In *Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE*, pages 13–16, Dept. of Electr. Eng., Arizona State Univ., Tempe, AZ, 2008.

[183] Wenping Wang, Shengqi Yang, Sarvesh Bhardwaj, Rakesh Vattikonda, Sarma Vrudhula, Frank Liu, and Yu Cao. The impact of NBTI on the performance of combinational and sequential circuits. In *Proceedings of the 44th annual Design Automation Conference*, DAC '07, pages 364–369, New York, NY, USA, 2007. ACM.

[184] Wenping Wang, Shengqi Yang, Sarvesh Bhardwaj, Sarma B. K. Vrudhula, Frank Liu, and Yu Cao. The Impact of NBTI Effect on Combinational Circuit: Modeling, Simulation, and Analysis. *IEEE Trans. VLSI Syst.*, 18(2):173–183, 2010.

[185] Yih Wang, Uddalak Bhattacharya, Fatih Hamzaoglu, Pramod Kolar, Yong-Gee Ng, Liqiong Wei, Ying Zhang, Kevin Zhang, and Mark Bohr. A 4.0 GHz 291Mb voltage-scalable SRAM design in 32nm high- metal-gate CMOS with integrated power management. In *ISSCC*, pages 456–457, 2009.

[186] Yu Wang, Xiaoming Chen, Wenping Wang, Varsha Balakrishnan, Yu Cao, Yuan Xie, and Huazhong Yang. On the efficacy of input Vector Control to mitigate NBTI effects and leakage power. In *Proceedings of the 2009 10th International Symposium on Quality of Electronic Design*, ISQED '09, pages 19–26, Washington, DC, USA, 2009. IEEE Computer Society.

[187] C. T. White and T. N. Todorov. Carbon nanotube as long ballistic conductors. *Nature*, 393:240–242, February 1998.

[188] Wikipedia. Noise margin, 2012.

[189] Steven J. E. Wilton and Norman P. Jouppi. CACTI: An Enhanced Cache Access and Cycle Time Model. *IEEE Journal of Solid-State Circuits*, 31:677–688, 1996.

[190] David Wolpert and Paul Ampadu. Normal and Reverse Temperature Dependence in Variation-Tolerant Nanoscale Systems with High-k Dielectrics and Metal Gates. In Maggie X. Cheng, editor, *NanoNet*, volume 3 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 14–18. Springer, 2008.

[191] David Wolpert, Bo Fu, and Paul Ampadu. Temperature-Aware Delay Borrowing for Energy-Efficient Low-Voltage Link Design. In *Proceedings of the 2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip*, NOCS '10, pages 107–114, Washington, DC, USA, 2010. IEEE Computer Society.

[192] Liang Xiaoyao and et al. Latency adaptation for multiported register files to mitigate the impact of process variations, 2006.

[193] Yun Ye, F. Liu, Min Chen, S. Nassif, and Yu Cao. Statistical Modeling and Simulation of Threshold Variation Under Random Dopant Fluctuations and Line-Edge Roughness. *IEEE Trans. Very Large Scale Integr. Syst.*, 19(6):987–996, June 2011.

[194] Hui Zhang, Varghese George, and Jan M. Rabaey. Low-Swing On-Chip Signaling Techniques: Effectiveness and Robustness. *IEEE TRANSACTIONS ON VLSI SYSTEMS*, 8(3):264–272, 2000.

[195] Yan Zhang, Dharmesh Parikh, Karthik Sankaranarayanan, Kevin Skadron, and Mircea Stan. HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects. Technical report, University of Virginia, 2003.

[196] W. Zhao and Y. Cao. New generation of predictive technology model for sub-45 nm design exploration. *IEEE Trans. Electron Devices*, 53(11):2816–2823, November 2006.

[197] Ying Zhou, Rouwaida Kanj, Kanak Agarwal, Zhuo Li, Rajiv Joshi, Sani Nassif, and Weiping Shi. The impact of BEOL lithography effects on the SRAM cell performance and yield. In *Proceedings of the 2009 10th International Symposium on Quality of Electronic Design*, ISQED '09, pages 607–612, Washington, DC, USA, 2009. IEEE Computer Society.

[198] James F. Ziegler. Terrestrial cosmic rays. *IBM Journal of Research and Development*, 40(1):19–40, 1996.

Appendix A

Our Published Paper (in ISQED–2012) [147]

VAR-TX: A Variability-Aware SRAM Model for Predicting the Optimum Architecture to achieve Minimum Access-Time for Yield Enhancement in Nano-scaled CMOS

Jeren Samandari-Rad[1], Matthew Guthaus[2], Richard Hughey[2]
[1]Department of Electrical Engineering UCSC, Santa Cruz, CA 95064 USA
[2] Department of Computer Engineering UCSC, Santa Cruz, CA 95064 USA
[1]jeffsrad@soe.ucsc.edu, mrg@soe.ucsc.edu, rph@soe.ucsc.edu

Abstract

In this paper we propose a new hybrid analytical-empirical model, called VAR-TX, that exhaustively computes and compares all feasible architectures subject to inter-die (D2D) and intra-die (WID) process variations (PV). Based on its computation, VAR-TX predicts the optimal architecture that provides minimum access-time and minimum access-time variation for yield enhancement in future 16-nm on-chip conventional six-transistor static random access memories (6T-SRAMs) of given input specifications. These specifications include SRAM size and shape, number of columns, and word-size. We compare the impact of D2D and WID variations on access-time for 16-nm SRAM with the 45-nm and 180-nm nodes and demonstrate that the drastic increase in the 1- and 3-sigma of the smaller nodes is mainly due to the increase in the WID variations. Finally, our model disputes previously published works—suggesting that square SRAM always produces minimum delays—and significantly extends and enhances the older models by adding both an extra dimension of architectural consideration and additional device parameter fluctuation to the analysis, while producing delay estimates within 4% of Hspice results.

Keywords

SRAM, variability, optimum architecture, access-time, yield

1. Introduction

Design variability due to D2D and WID process variations has the potential to significantly reduce the maximum operating frequency and the effective yield of high-performance chips in future process technology generations. This variability manifests itself by increasing the access-time variance and mean of fabricated chips.

As device feature size is continually reduced in the semiconductor industry, the impact of fabrication variability on product reliability, yield and cost is dramatically increased. Mismatched MOS transistors impact the performance of today's scaled technology more than ever because both device dimensions and available signal swing are significantly reduced. Embedded microprocessors and other high-performance on-chip modules incorporate SRAM or cache components that play significant roles in overall chip functionality and reliability. Unwanted variations in SRAM circuits may result in access-time variations and chip functional failures. Therefore, comprehensive modeling and optimization of all possible architectures and organizations, including analysis of device parameter variations, is essential in confronting the ever-growing scaling issues.

The performance (and therefore, cost) of a given on-chip SRAM cannot be assessed adequately without investigating different silicon alternatives. For example, one cannot compare two different SRAM organizations without considering differences in access or cycle times. Similarly, one must account for chip area and power requirements to achieve an optimal design.

Many modeling techniques have been proposed to minimize the impact of process variations. In the SRAM and cache field, chip-area models [1], power/leakage models [1, 2, 17], access-time models [3, 13, 14], and failure probability models [13, 14] have been published. Further, there are newer techniques that can be used to combat process variations more effectively such as adaptive body biasing (ABB) [10] or chip-by-chip resource resizing in various micro-architectural structures [11]. However, they have inherent costs, must be applied with great caution, or require modifications to the chip architecture. Such costly complications demonstrate the importance of inexpensive and early modeling to determine the optimal design of future systems that will allow SRAM to be more tolerant to process variations.

In two recent models [9, 14], path-based variation-induced statistical timing analyses of SRAM memories were proposed. Although insightful, neither of these or other subsequent approaches capture the *architectural dependence* of the gate delay due to variability of fan-out gates; nor do they address the WID and D2D variability of *Vdd* (which we confirm is not as significant as threshold and transistor length). The former case, in particular, is important in selecting the architecture that reduces both the delay and the delay variation and hence increases the yield while meeting given area and power constraints. Rather than modeling the *Vdd* variation as a static IR drop or dynamic $L\ di/dt$, we chose to model it in the same way we model the length, but with only half the variance of L. We do this because we are interested in capturing the effect of *Vdd* variation on delay only, and not on leakage current or power. In this paper, therefore, we propose a new path-based approach to statistical timing analysis that considers both the architecture- and process-variations.

We model variations of the gate delay due to fluctuations of the input slope and output loads resulting from variations

of fan-in and fan-out stages in the path for all possible 6T-SRAM architectures. We propose a model where the D2D & architecture-dependent WID variations of all the major parameters of the device are modeled as two separate components. Furthermore, we propose efficient methods for computing path delay variability due to either source, as well as their combined effect.

Specifically, this paper makes three major contributions:

- We propose a novel hybrid analytical-empirical model that exhaustively computes and compares the sensitivity of different 6T-SRAM architectures to the variations in threshold voltage (*Vth*), gate length (*L*), and supply voltage (Vdd). This enables the user to select the optimal architecture that gives the minimum delay and/or minimum delay variation while providing the maximum yield possible, for the given area and power constraints. In considering the sensitivity of the critical path to variations in both the overall architecture and within the individual devices, we not only add a new dimension to the path-based statistical timing analysis but also significantly improve upon the previous access-times models [8, 9, 13, 14]—which neither considered architectural sensitivity nor all three parameter variations.
- Using our model, we dispute previously published works that suggest square SRAM always produce minimum delays. We show that minimum access-time and/or access-time variation can be obtained from a non-square SRAM.
- Additionally, we present the access-time variation calculated by our model for the future 16-nm node and compare it to those of the recent 45-nm and older 180-nm to show the larger impact of process variations in increasingly small devices and therefore help shed light on the challenges of future robust circuit design.

For simplicity and focus purposes, we ignore the temperature dependency of delay because, as opposed to its significant influence on leakage power, the impact of temperature (in the range of our interest for SRAM namely ~50 to ~100°C) on gate delay and on wire delay are sufficiently small [14] and do not change our overall results for first order approximation significantly. This is because, unlike multicore chips, SRAM chips have very regular and dense structure and do not have many interconnects.

However, we take the effect of Negative Bias Temperature Instability (NBTI) and Hot Carrier Injection (HCI) on the delay of 16-nm into consideration, by using the future *Vth* variability forecast by International Technology Roadmap for Semiconductor (ITRS) and ASU Predictive Technology Model (PTM) [15, 16]. NBTI and HCI effects increase the *Vth* of new technologies (32 nm and shorter nominal channel lengths), considerably, especially over time.

The proposed model and analysis method was applied to standard 6T-SRAM, although extensions to other types of memory such as 8T-, 10T-, or multi-ported SRAM, cache and CAM are straightforward.

Our hybrid analytical-empirical model was partially built on the empirical data collected from the results of numerous restricted simulations on SRAMs composed of the latest complex circuits. All circuits were designed *at the transistor level,* with each transistor in the circuit subject to WID and D2D variations in *Vth, L,* and *Vdd*. Our model also includes *layout parasitics* (e.g., the resistance and capacitance of all the bitlines (wires) and wordlines (wires) in the 6T-cell array). We verify the accuracy of our proposed model assumptions through Monte Carlo simulations, and validate our model optimization capability by comparing our access-time results with that obtained by Mukhopadhyay and VARIUS [13, 14].

Section 2 of this paper briefly reviews 6T-cell design challenges and the main causes for failure. Section 3 outlines our VAR-TX model. Section 4 presents and analyzes our results. Finally, Section 5 summarizes our findings.

2. SRAM overview

The six-transistor-cell static random access memory (6T-SRAM) is the conventional choice for most on-chip memory designs. With power applied, SRAM provides permanent data storage. Fig.1 shows the schematic for the 6T cell of a 6T-SRAM. Cell design requires a complex balancing among several factors including speed, silicon area, and power/leakage consumption [6, 7, 17]. The balancing task is challenging due to conflicting interactions among several factors.

For example, to maintain cell stability and good soft-error (transient errors induced by radiation) immunity [4] while keeping access-time short, one might specify large transistor sizes. But large transistors occupy more area and result in increased leakage. Similarly, improving static noise margin (SNM) with smaller pass transistors can lead to a worse write margin [6]. Transistor sizing and circuit styles for 6T-SRAM components (decoders, sense amps, etc.)—and the interconnect sizing, buffers, and SRAM array partitioning—must all be traded off against delay, area, and power consumption.

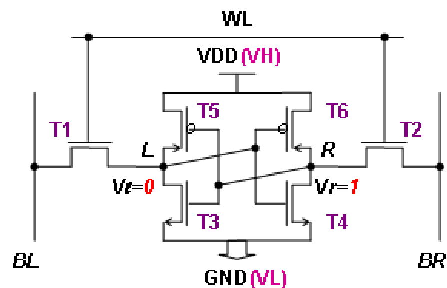


Figure 1: 6 transistor (6T) storage cell.

2.1. Failure in SRAM

As Fig.1 shows, the 6T-SRAM cell consists of two N-type access transistors and two cross-coupled CMOS

inverters. Large mismatches in transistor strengths due to scaling (or fluctuations in die electrical characteristics) can cause the cell to fail.

SRAM cell failures can be classified into four categories: *read*, *write*, *access-time,* and *hold* failures.

Read Failure can be defined as flipping data during an SRAM cell read. Because a voltage divider exists between BL (precharged at VDD) and GND, $V\ell$ (the node *L* voltage, a 0 in Fig.1) is raised from zero to Vread through T1 and T3. If Vread exceeds the tripling voltage, Vtrip, of the {T4, T6} inverter, the cell state flips—a read failure. Fluctuations in the *Vth* of T1 and T3 (or of the T4/T6 *Vth*) lead to large variations in Vread (or Vtrip, respectively) [5].

Write Failure occurs when a memory cell does not register an input change correctly. Because a voltage divider exists between BR (at GND) and VDD, Vr transitions to Vwrite through T2 and T6 when a zero is written to node R in place of an original one. The write fails if Vwrite is larger than the Vtrip of the {T3, T5} inverter. *Vth* variations in T2 and T6, and also in T1 and T5—typically the smallest transistors in the cell—cause large variations in Vwrite. This Vwrite ambiguity means a high write-failure probability [5].

Access Time Failure: The Access Time of a cell (Taccess) can be defined as the time required to develop a predefined voltage between BL and BR. When node *L* stores a zero, BL will discharge through T1 and T3 in a read operation. The T1 and T3 strengths influence the discharge speed. *Vth* variations in these transistors cause a spread in Taccess [5]. If Taccess exceeds the maximum tolerable limit (Tlimit), an access failure is the result.

Hold Failure: The destruction of the cell content in standby mode with the application of a lower supply voltage VH (below 0.5V in our 16-nm node, primarily to reduce leakage in standby mode) is known as hold failure [13].

Read, write, access-time, and hold failure probabilities, which are highly sensitive to *Vth* variation [5] and considerably sensitive to *L* and *Vdd* variations [13], can be as high 5×10^{-3} for the 16-nm process.

3. Our proposed model

Among the four types of SRAM failures (Read failure, Write failure, Access failure, and Hold failure), Access Failure is by far the most influential culprit for chip failure [13]; therefore, we only consider Access Failure in our analysis of SRAM delay and delay variations in this paper. However, the minimum clock period that we choose for each technology node is determined in such a way to avoid all four types of SRAM cell failures (*read*, *write*, *access-time,* and *hold* failures) up to the failure probability specified by the ITRS [15], as discussed in section 2.1. The clock period is the summation of precharge time, read/write time, and senseamp and output-driver component delays managed by internal-clock circuitry—which sets the slew rate and pulse width for the timing of SRAM components. These timings are empirically tested, adjusted, and verified to ensure sufficient SNM and stability and therefore robust memory design.

Model stage delays depend on input slopes and output loading. This means different SRAM architectures/ organizations exhibit different component delays. For example, assuming all other parameters are equal, the precharge component delay in an architecture that uses a larger row decoder, and therefore longer bitlines, exceeds the precharge delay in an architecture that uses a smaller row decoder and shorter bitlines. This is because the input rise time and output loading of the precharge in the former are larger than in the latter. Of course, this is only a valid statement if all other parameters, such as word-size, are the same for the two architectures.

We measured and recorded the delay and delay variation for each stage of each SRAM component. The stage delays and variations were then combined to calculate total component delays and total component delay variations. This section details how we measured and combined the delays, and how these delays are used to obtain access-times and variations of access time. Lastly, we explain our new VAR-TX model.

3.1. Derivation of access-time and its variation

The proposed model and analysis method was applied to the variation in all three major device parameters and for all different feasible 6T-SRAM architectures. We obtained D2D and WID device parameter variation from predictions of the International Technology Roadmap for Semiconductors [15] and the experimental data of other published works [14]. For spatial correlation component allocations, we used our own empirically collected data.

To compute the WID path delay component of process variability, we first compute the sensitivity of gate delay, output slope, and input load with respect to the input slope, output load and device parameters for all feasible architectures. Using these sensitivities, we then express the path delay variation as an analytical expression of the device parameter variation, allowing for very efficient analysis of WID variability, including an accurate model for spatial correlation. Since the D2D component of path delay variability is dependent on a single random variable, we can compute it efficiently through enumeration of its probability distribution. We then compute the joint path delay distribution through the convolution of WID and D2D delay distribution components to obtain the distribution of the total delay variability.

In order to extract the total delay variation due to the device parameter variation, we use a first-order approximation which is widely used in statistical timing analysis [8, 9, 12], shown in (1).

$$P_{total,\,i} = P_0 + \Delta P_{D2D} + \Delta P_{WID,\,i} = P_{D2D} + \Delta P_{WID,\,i} \quad (1)$$

where *P* represents any of the three parameters in the system, such as *Vth*. We model each device parameter $P_{total,,i}$ of device *i* as the algebraic sum of a D2D device parameter

P_{D2D} and WID device parameter variation $\Delta P_{WID,i}$. The D2D device parameter is defined as $P_{D2D}=P_o+\Delta P_{D2D}$, where P_o is the nominal value of P, and ΔP_{D2D} is the change in the delay of device due to D2D variation of the parameter P.

We can apply this generic equation to the specific device parameters that we consider in our model—Vth, L, and Vdd:

$$Vth_{total,\,i} = Vth_{D2D} + \Delta Vth_{WID,\,i} \quad (2)$$

$$L_{total,\,i} = L_{D2D} + \Delta L_{WID,\,i} \quad (3)$$

$$Vdd_{total,\,i} = Vdd_{D2D} + \Delta Vdd_{WID,\,i} \quad (4)$$

For each device parameter P, all devices on a die share one variable P_{D2D} for the D2D component of their $P_{total,i}$, which represents the *due-to-P* mean of the gate of a particular die (e.g., Vth_{D2D}, L_{D2D}, and Vdd_{D2D} represent the mean of all devices on a die with respect to Vth, L, and Vdd, respectively). The WID component of each device has a separate independent random variable $\Delta P_{WID,i}$, where all random variables $\Delta P_{WID,i}$ have identical probability distributions (e.g., each device on a die has a separate independent random variables $\Delta Vth_{WID,i}$, $\Delta L_{WID,i}$, and $\Delta Vdd_{WID,i}$ that are different than those of the neighboring devices). The D2D variation P_{D2D} has a mean which is equal to the nominal value of the P of device. The WID variation $\Delta P_{WID,i}$ has systematic and random components of which the latter has a mean of zero. The total variation $P_{total,}$ therefore, has a mean equal to sum of the mean of P_{D2D} and mean of $\Delta P_{WID,i.}$ We assumed that all three random variables $P_{total,i}$, P_{D2D}, *and* $\Delta P_{WID,i}$ have a normal distribution, which is a common assumption since device threshold voltage, length, and supply voltage are physical quantities.

We obtain the distribution of the path delay D_p resulting from the variation of all device parameters and delay of the individual gates in the path of a certain architecture through Eq.(5)—where the path delay D_p is a random variable, and D_i is the delay of gate i as a function of its device parameters and the sum is taken over all gates of a path of certain architecture.

$$D_p = \sum_i^n D_i(P_{D2D} + \Delta P_{WID,\,i}) \quad (5)$$

The computation of the D_p distribution is difficult since D_i is a non-linear function that cannot be accurately expressed in closed form. Therefore, we resort to two feasible methods. One method for computing the distribution of D_p is through Monte-Carlo simulation that we perform in section 4.1. Another method is to use the following simplifying assumption:

$$D_i(P_{D2D} + \Delta P_{WID,\,i}) = D_i(P_{D2D}) + \Delta D_i(\Delta P_{WID,\,i}) \quad (6)$$

which shows that the gate delay in a certain architecture is approximated by the sum of the delay of the D2D and variation of WID of the gate in that architecture. The assumption of Eq.(6) allows us to compute $D_i(P_{D2D})$ and $\Delta D_i(\Delta P_{WID,,i})$ independently and then combine them to obtain the total path delay distribution D_p, as follows:

$$D_p = \sum_i^n D_i(P_{D2D}) + \sum_i^n \Delta D_i(\Delta P_{WID,\,i}) \quad (7)$$

We discuss the computation of the two components of D_p in the following two sub-sections.

3.1.1. D2D variability analysis

To compute the delay due to D2D variation we need to compute $D_{p,D2D}$ as function of D2D device parameters as in Eq.(8).

$$D_{p,\,D2D} = \sum_i^n D_i(P_{D2D}) \quad (8)$$

$$= f[\sum_i^n D_i(Vth_{D2D}), \sum_i^n D_i(L_{D2D}), \sum_i^n D_i(Vdd_{D2D})]$$

For each parameter (Vth, L, Vdd), the corresponding gate delay in D_p,$_{D2D}$ shares a single random variable, therefore, the D2D variation of Dp due to each parameter can be computed separately through enumeration of the distribution of Vth, L, and Vdd (Vth_{D2D}, L_{D2D}, Vdd_{D2D}). We enumerate the different possibilities from the worst case to the best case process corners for each of the three parameters, and compute the resulting path delay $D_{p,D2D}$ for each of the three cases individually. The probability distribution of $D_{P,D2D}$ for each individual parameter is then computed by considering the probability distribution (Vth_{D2D}, L_{D2D}, or Vdd_{D2D}) of the selected device parameter (Vth, L, or Vdd) and their associated resulting path delay for each enumeration.

We then combine the mean and the variance of all three normal distributions to obtain the mean and variance of $D_{p,D2D}$. In our experiments, discretization of Vth_{D2D} into 30 device thresholds and L_{D2D} into 20 device lengths and Vdd_{D2D} into 3 device supply voltages was sufficient to obtain a high level of accuracy. This requires simulating each path 30 times for Vth, 20 times for L, and 3 times for Vdd, for each of the feasible architectures, which is a relatively low cost for computing $D_{p,D2D.}$

3.1.2. WID variability analysis

The path delay variation due to WID device parameter variation (the second term in Eq.(6)) is a function of multiple independent random variables, which requires an impractical number of simulations for computing $D_{p,WID.}$ Therefore, we make a second simplifying assumption, namely that $\Delta D_i(\Delta P_{WID,i})$ can be approximated linearly as

$$\Delta D_i(\Delta P_{WID,\,i}) = \frac{\partial D_i}{\partial P_{WID,\,i}} \times \Delta P_{WID,\,i} = coef_i \times \Delta P_{WID,\,i} \quad (9)$$

for small values of $P_{WID,i}$, where the sensitivity of the delay with respect to device parameter $\partial D_i/\partial P_{WID,i}$ is computed at the nominal device parameter value. The simplification of Eq.(9) allows us to compute the change of path delay $D_{p,WID}$ due to WID device parameter variation analytically and efficiently, using pre-computed delay sensitivities ($coef_i$).

When computing $D_{p,WID}$, the dependence of the delay of gate i on gate input load of its fan-out gate i+1 must be considered, which is a function of the device parameter $\Delta P_{WID,i+1}$. Similarly, the delay of gate i is dependent on its input slope, which is a function of all device parameters $\Delta P_{WID,j}$, where gate $j < i$ precedes gate i in the path. We therefore extend the linear assumption of Eq.(9) to the change of a gate delay and output slope due to input slope and output load and formulate the computation of $D_{p,WID}$ for each parameter *Vth, L,* and *Vdd* in the same way, which is shown below for the threshold voltage case.

The change in path delay due to *Vth* ($D_{p,WID,Vth}$) is the sum of the individual gate delay changes ΔD_i due-to-*Vth*, where each of the gate delay changes and their corresponding output slope changes are a function of the change in output slope of the *preceding* gate ΔS_{i-1}, the change in input load of the *succeeding* gate ΔCl_{i+1}, and the WID device threshold $\Delta Vth_{WID,i}$:

$$\Delta D_i = f(\Delta S_{i-1}, \Delta Cl_{i+1}, \Delta Vth_{WID,\, i}) \qquad (10)$$

$$\Delta S_i = f(\Delta S_{i-1}, \Delta Cl_{i+1}, \Delta Vth_{WID,\, i}) \qquad (11)$$

The change in delay, slope, and input capacitance of a single gate is approximated as a sum of products of the sensitivities and the change in the threshold values:

$$\Delta D_i = \frac{\partial D_i}{\partial S_{i-1}} \times \Delta S_{i-1} + \frac{\partial D_i}{\partial Cl_{i+1}} \times \Delta Cl_{i+1} + \frac{\partial D_i}{\partial Vth_i} \times \Delta Vth_{WID,\, i} \quad (12)$$

$$\Delta S_i = \frac{\partial S_i}{\partial S_{i-1}} \times \Delta S_{i-1} + \frac{\partial S_i}{\partial Cl_{i+1}} \times \Delta Cl_{i+1} + \frac{\partial S_i}{\partial Vth_i} \times \Delta Vth_{WID,\, i} \quad (13)$$

$$\Delta Cl_i = \frac{\partial Cl_i}{\partial Vth_i} \times \Delta Vth_{WID,\, i} \qquad (14)$$

The seven basic sensitivities of delay (ΔD_i) and slope (ΔS_i) with respect to input slope, output load and device threshold and the sensitivity of gate input load (ΔCl_i) with respect to device threshold are pre-computed for each gate over a range of output load and input slope conditions.

In this paper, we computed the sensitivities for all cases of *Vth*, *L*, and *Vdd* during circuit simulation by use of the curve fitting method illustrated for *Vth* in Fig.2. The delay for a gate (located on the critical path) with respect to either of the device parameters (e.g., $d_{g,WID,Vth}$) is the summation of the gate nominal delay (d_{go}) and the additional delay caused by parameter fluctuation of the device (e.g., $\Delta d_{go,Vth}$).

$$d_{g,WID,\, Vth} = d_{g0} + \Delta d_{g0,Vth} \qquad (15)$$

For small scale fluctuations, such first-order linear approximation is accurate enough. Fig.2 shows the delay and gate threshold fitting curve for the computation of the sensitivity $\partial D_i / \partial Vth_{WID,i}$ (represented by three different slopes a, a', a'' for three different architectures in 16-nm 64KB SRAM), and the linear fits match the Hspice simulation for the range under consideration. A similar agreement holds for *L* and *Vdd* cases in different architectures, as well. These basic sensitivities along with their associated WID variations ΔP_{WID} (discussed below in this section) are stored in tables and are accessed during the computation of $D_{p,WID}$ for a particular path using linear interpolation of the stored values in the table.

It is interesting to observe in Fig.2 how the delay and gate threshold fitting curves for the same gate can be different under different circumstances. While the upper curve (line 1) shows a larger slope (indicating larger variation) for a gate used in an 1:64:1024 (columns:word-size:rows) architecture, the bottom curve (line 3) exhibits a relatively smaller slope (indicating smaller variation) for the same gate used in a 64:64:16 architecture. The middle curve (line 2) of architecture 4:64:256 shows a slope between those of the other two. Such differences in the slopes is mainly due to different cumulative loading effect of the preceding gates on the input slope of the gate and the different loading effect of the succeeding gate on the output capacitance of the gate in different architectures.

The access-time is the summation of the associated critical path nominal delay (*D*o) and the additional delay caused by parameter fluctuations of each device on the path, assuming *n* total devices. Since the numerical value of *n* and each of the *n* gates' total parameter variations in different architectures are different, the access-time and variation of access-time for different architectures are different. In sections 4.2 and 4.3 we show how choosing the optimal architecture can reduce the access-time and/or variation of access-time.

For WID variations ΔP_{WID}, there are both correlated (systematic) and random components. To capture this effect, we use the method introduced by Agarwal [8]. The SRAM area is divided into a multi-level quad-tree partitioning as shown in Fig.3. For each quadrant, we generate a random variable according to a normal distribution. The ΔL_{WID} of a transistor can then be obtained by adding up all the random variables of the quadrants it belongs to. For example, the transistors in Gate1 and Gate2, shown in Fig.3, have WID gate length variations of $\Delta L_{WID,Gate1} = rand_{0;1} + rand_{1;1} + rand_{2;1}$ and $\Delta L_{WID,Gate2} = rand_{0;1} + rand_{1;1} + rand_{2;4}$, respectively. Therefore, transistors in Gate1 and Gate2 have strong correlation because they share the variables $rand_{0;1}$ and $rand_{1;1}$, while transistors in Gate1 have less correlation with transistors in Gate3 because they only share the variable $rand_{0;1}$. We apply the same procedure for WID of *Vdd* (ΔVdd_{WID}) and the systematic-WID of threshold ($\Delta Vth_{WID,sys}$). Our $\Delta Vth_{WID,sys}$ has an inverse relation to the squareroot of $L \times W$ [13, 19]. We model the random-WID of *Vth* ($\Delta Vth_{WID,rand}$) as a random variable which obeys a normal distribution [13] only due to random dopant effect (RDF) and line edge roughness (LER) [19]. The summation of $\Delta Vth_{WID,sys}$ and $\Delta Vth_{WID,rand}$ gives ΔVth_{WID}. To model *L* and *Vdd* to exhibit highly correlated variation and *Vth* to exhibit mostly random (weakly correlated variation), we allocate most of ΔL_{WID} and ΔVdd_{WID} to the higher levels and most of $\Delta Vth_{WID,sys}$ to lower levels in the hierarchy of Fig.3.

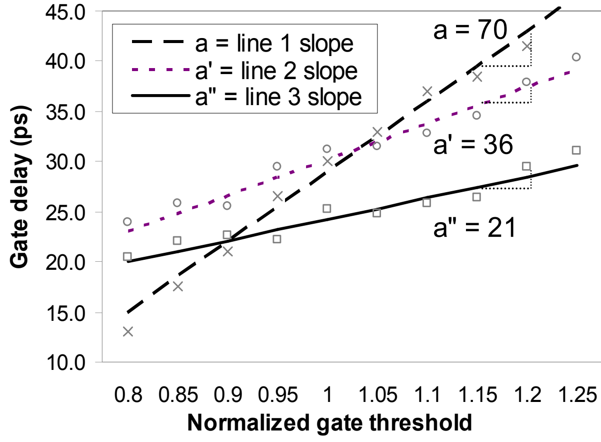


Figure 2: Curve fitting for Hspice simulation for an SRAM.

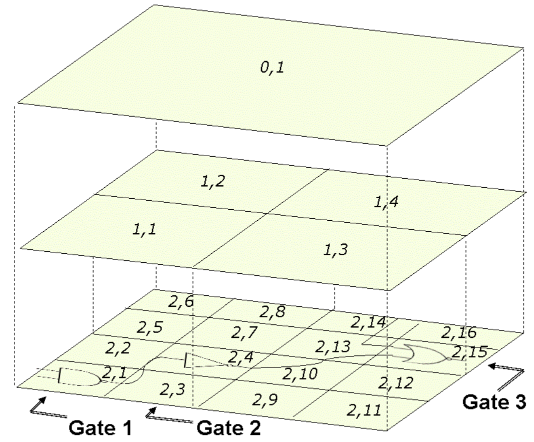


Figure 3: Spatial correlation modeling for WID variations (Based on Fig.1 of Agarwal [8]).

We apply 6 levels of quadrants with the top quadrant the entire SRAM and the bottom quadrant the individual devices for gate parameter modeling.

We then combine Equations (12)-(14) to obtain an expression of ΔD_i as a function of basic sensitivities and WID device threshold variations. The delay change coefficients of this function are efficiently computed for all gates in the path using a single traversal of the path for each architecture using the basic seven sensitivities. We then collect all coefficients of gate delays with respect to each WID device threshold and express the total change in path delay $D_{p,WID,Vth}$ (*due to Vth*) as follows:

$$D_{p,WID,\ Vth} = \sum_{i}^{n} x_i \times \Delta Vth_{WID,\ i} \qquad (16)$$

where x_i is the coefficient of total path delay change due to WID device threshold ΔVth_i at gate i. Eq. (17) shows the total WID path delay change *due-to-all-device-parameters* for one of the m number of architectures.

$$D_{p,WID} = \sum_{i}^{n} (x_i \times \Delta Vth_{WID,\ i} + y_i \times \Delta L_{WID,\ i} + z_i \times \Delta Vdd_{WID,\ i}) \qquad (17)$$

Given the mean (μVth_i, μL_i, and μVdd_i) and the standard deviation (σVth_i, σL_i, and σVdd_i) for the WID device threshold ΔVth_i, WID device length ΔL_i, and WID device supply voltage ΔVdd_i, with normal distribution, and the coefficients x_i, y_i, and z_i, we can compute the mean and standard deviation of the probability distribution for $D_{p,WID}$ directly using the following standard equations:

$$\mu D_{p,WID} = \sum_{i}^{n} (x_i \times \mu Vth_i + y_i \times \mu L_i + z_i \times \mu Vdd_i) \qquad (18)$$

$$\sigma D_{p,WID} = \sqrt{\sum_{i=1}^{n} (x_i^2 \times \sigma Vth_i^2 + y_i^2 \times \sigma L_i^2 + z_i^2 \times \sigma Vdd_i^2)} \qquad (19)$$

Given pre-characterized sensitivities, the final computation of the distribution of $D_{p,WID}$ is performed very efficiently and requires only a single traversal of the path for each of the architectures. We show the major impact of the architecture-dependent WID variations on the access-time by comparing its distribution ($D_{p,WID}$), computed through the proposed analytical approach, with that of the total D_p obtained through the proposed analytical approach and compared against Monte Carlo simulation in Section 4.1.

3.1.3. Combined WID and D2D analysis

After computing the two components of path delay variation, $D_{p,D2D}(P_{D2D})$ and $D_{p,WID}(\Delta P_{WID,i})$, we compute the distribution of the total path delay D_p. Since P_{D2D} and $\Delta P_{WID,i}$ are independent random variables, this involves the convolution of the two distributions. However, since $D_{p,D2D}$ is not normal, the convolution can not be performed analytically, and must be done by discretizing the two distributions and then taking their convolution numerically. By repeating the same procedure used for the computation of D_p, we find the total path delays D'_p, D''_p, D'''_p … for all possible architectures, verify them by Monte Carlo simulation, and store them in tables. A Monte Carlo verification sample is shown in section 4.1.

3.2. Model assumptions and implementation

Although labor-intensive (mainly during the data collection for sensitivities step), the construction of a hybrid analytical-empirical model such as this one takes a reasonable time on a small cluster (weeks, not months). The initial expensive sensitivity analysis characterization involved in the flow is compensated for by the time savings

in the subsequent short run-times. While Hspice Monte-Carlo simulations for each of the many possible configurations of the actual large SRAM circuits can take days (which makes such alternatives comparatively quite expensive), VAR-TX carries out the same analysis in minutes. Despite the time savings, for the circuits we have chosen, our model produces delay estimates within 4% of Hspice results.

A total independent variation of 8.8% for the WID sigma of *Vth*, 4.4% for Lgate, and 2% for *Vdd* were assumed for our variability analysis of 16-nm node. For D2D independent variance we assumed 4% for either of *Vth* and *L*, and 2% for *Vdd*. We chose these percentages based on the manufacturing process variation forecast of ITRS [15]. Our simulations are based on ASU Predictive Technology Models (PTM) [16]. Sixty different transistor models, each with a different value for V_{TH0}, were used to model *Vth* variations for our SRAM circuits. To model gate length variations, we stipulated 20 different values of deviation from the standard minimum-size transistor length. Finally, we modeled *Vdd* variations using two extreme cases: the default supply voltage plus 1-sigma and the default *Vdd* less 1-sigma.

Every transistor in the netlist was subject to both random and spatially-correlated systematic fluctuations of *Vth*, *L*, and *Vdd*. The proposed model assumptions are verified through Monte Carlo simulation and validated through comparison with VARIUS [14], which show that the proposed approach produces very accurate results.

3.3. Model optimization

In addition to computing the access-time of a given SRAM system, VAR-TX performs exhaustive computations and comparisons based on the user entry (i.e. SRAM size, word-size), using its embedded library of lookup tables (constructed from the linearized device delays for different configurations) to provide the minimum-access-time architecture/organization that satisfies a given desired power and area requirement from the modeled alternatives. VAR-TX does this within thirty seconds, even for large SRAM circuits with nearly countless critical parameter fluctuations. VAR-TX also provides a measure of the expected variability in this minimum access-time.

4. Results and analysis

We used the mixed-signal Ultrasim simulator (MMSIM72-Ultrasim64, Cadence Inc.) to produce the results presented in this section.

4.1. Verification by Monte-Carlo

To validate the accuracy of the proposed approach, we compare the distribution of *Dp*, Eq.(6), computed through the proposed analytical approach (section 3.1.3) with that obtained through Monte Carlo simulation in this section. For each transistor, we model each of the gate parameters as:

$$P = P_0 + \Delta P = P_0 + \Delta P_{D2D} + \Delta P_{WID} \quad (20)$$

where P_0 is the nominal value representing either Vth_0, L_0, or Vdd_0. For each of the gate parameter variations ΔP (*e.g.*, ΔVth_k, ΔL_k, and ΔVdd_k, used in Eqs.(22 to 24)) there are D2D and WID components ΔP_{D2D} and ΔP_{WID}, respectively. For D2D variations, we generate a random variable for each of *Vth*, *L*, and *Vdd* for each chip according to a normal distribution. For WID variations, there are both correlated (systematic) and random components. To capture this effect, we use the same multi-level quad-tree method discussed in section 3.1.2 and shown in Fig.3. For each quadrant we generate a random variable according to a normal distribution. The ΔL_{WID} and ΔVdd_{WID} and $\Delta Vth_{WID,sys}$ of a transistor can be obtained by adding up all the random variables of the quadrants that *Vth*, *L*, or *Vdd* belongs to. $\Delta Vth_{WID,rand}$ is obtained from a random variable which obeys a normal distribution that represents the random-WID component of *Vth*. ΔVth_{WID} is obtained by adding up $\Delta Vth_{WID,sys}$ and $\Delta Vth_{WID,rand}$. Similar to section 3.1.2, transistors of Gate1 and transistors of Gate2 have strong correlation because they share the variable $rand_{0;1}$ and $rand_{1;1}$, while transistors in Gate1 have less correlation with transistors in Gate3, because they only share the variable $rand_{0;1}$. The summation ΔP_{D2D}+ΔP_{WID} for each of *Vth*, *L*, and *Vdd* (assigned to each device separately) gives ΔP for every parameter of each device.

We use Monte-Carlo simulation to generate all the random variables necessary in the model and generate ΔP_{D2D} and ΔP_{WID} for the gate threshold voltage, length, and supply voltage for each device on the delay path in the 16-nm, 45-nm, and 180-nm 64KB 6T-SRAM. For gate parameter modeling, we apply 6 levels of quadrants (sufficiently fine partitioning for our first order analysis) with the top quadrant the entire SRAM and the bottom quadrant the devices. We then use the fitting curve introduced in Section 3.1.2 to obtain the coefficients a_k, b_k, and c_k of the combined WID+D2D gate delay changes ΔP to calculate the change-in-delay of every gate *k* on the critical path (ΔD_{Pk}), resulting from the change in parameter *P* of gate *k*, (e.g., ΔD_{Vthk}, ΔD_{Lk}, and ΔD_{Vddk}) and, subsequently, compute the delay of all the possible paths D_{path} in the SRAM.

$$D_{path} = D_0 + \Delta D_{Vth\ i} + ... + \Delta D_{Vth\ n} \quad (21)$$
$$+ \Delta D_{L\ i} + ... + \Delta D_{L\ n} + \Delta D_{Vdd\ i} + ... + \Delta D_{Vdd\ n}$$

$$\Delta D_{Vth_k} = \partial D / \partial Vth_k \times \Delta Vth_k = a_k \times \Delta Vth_k \quad (22)$$

$$\Delta D_{L_k} = \partial D / \partial L_k \times \Delta L_k = b_k \times \Delta L_k \quad (23)$$

$$\Delta D_{Vdd_k} = \partial D / \partial Vdd_k \times \Delta Vdd_k = c_k \times \Delta Vdd_k \quad (24)$$

The delay for a critical path D_{path}, Eq.(21), with variation is the summation of the path nominal delay (D_o) and the additional delay caused by parameter fluctuation of each device on the path, assuming *n* total devices. We simulate 2000 chips, which is sufficient for our statistical analysis. The plots shown in Fig.4 show a close match between our

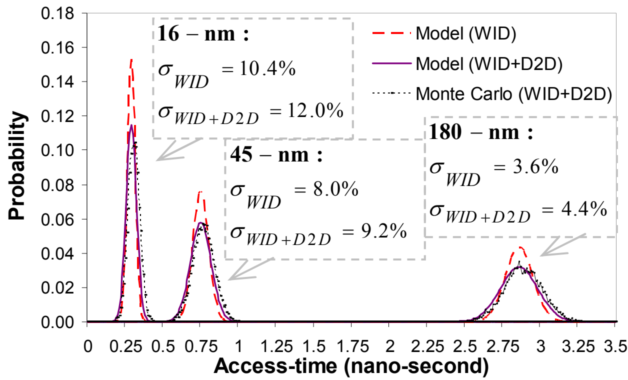


Figure 4: Verifying the proposed model with Monte Carlo.

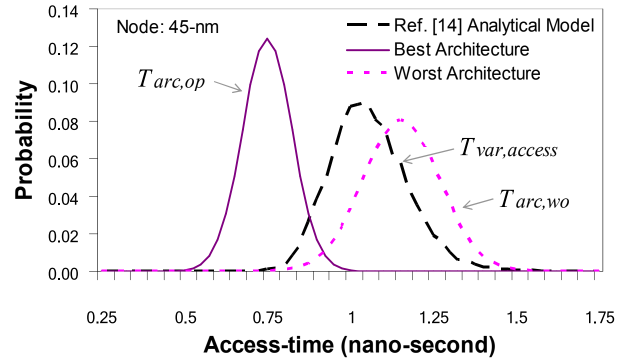


Figure 5: Validating optimization capability of our model.

proposed hybrid analytical-empirical approach and the Monte Carlo simulations.

There are two additional observations that can be made from Fig.4.

1) Most of the access-time variation is due to WID variations of the gates on the critical path, as the standard deviation (not the mean) of each of the dashed (red) curves—representing the distribution of the access-time due to cumulative WID fluctuation of the parameters that are denoted by σ_{WID} in the figure—is very close to the corresponding standard deviation of the solid (violet) curves—representing the distribution of the access-time due to combined cumulative WID+D2D fluctuation of the parameters that are denoted by $\sigma_{WID+D2D}$ in the figure.

2) Comparing the much higher 3-sigma deviation from the mean of delay curve of 16-nm with those of 45-nm and 180-nm, it is not unreasonable to anticipate that essentially a generation of performance gain could be lost in the upcoming 16-nm technology node unless new innovations in manufacturing process controls (e.g., an improved replacement for hafnium oxide, double/triple patterning technologies, EUV lithography, E-beam direct-write, and other maskless lithography), and new circuit design methodologies (e.g., row or column redundancy) are thoroughly investigated and effectively employed.

4.2. Validation of model optimization

To quantify the access-time improvement of our proposed approach, we compare the probability density function (PDF) of our optimal access-time $T_{arc,op}$ with both the PDF of our worst access-time $T_{arc,wo}$ and the PDF used in VARIUS [14], $T_{var,access}$—building on the work Mukhopadhyay [13], that uses $T_{var,access}$ α $(1/I_{dsatT1})$—for a given 45-nm 64KB 6T-SRAM in Fig.5. The mean and variance used in VARIUS [14] are very similar to the mean and variance of our worst case scenario ($T_{arc,wo}$), and are both considerably different from our calculated best case scenario, which clearly confirms the optimization capability of VAR-TX. Fig.5 illustrates that by choosing the optimum architecture in an SRAM design the access-time (and therefore the yield) can be improved by up to about 31%, with respect to prior models such as those proposed by Mukhopadhyay and VARIUS [13, 14].

Table 1 compares the mean, sigma, and 3 sigma of the access-time of the optimum, worst, and three other architectures that fall between the optimum and worst architectures against the access-time of VARIUS [14]. The drastic difference between the mean, sigma, and 3 sigma of the worst and optimum cases clearly emphasize the crucial role of selecting an optimum architecture in frequency improvement.

Table 1: Comparison of different architectures with VARIUS [14].

Architecture /Design selection	No. of gates in path	Mean access-time		Standard deviation		3 sigma delay	
		ns	% imp	ns	% imp	ns	% imp
Arc 1	25	0.32	24%	0.038	30%	0.435	25%
Arc 2	29	0.39	7%	0.047	14%	0.530	9%
Arc 3	33	0.48	-14%	0.059	-8%	0.657	-13%
Ref [14]	(?)	0.42	0%	0.055	0%	0.584	0%
Arc Worst	38	0.54	-29%	0.065	-19%	0.735	-26%
Arc Optim	27	0.29	31%	0.035	36%	0.394	32%

4.3. Access-time

We characterize our access-time results with the following terms:

ACS: minimum access-time for an SRAM where the optimal organization takes a square shape.

ACI: similar to ACS, but the optimal organization need not take a square shape. ACI is always smaller than or equal to ACS.

The two upper curves in Fig.6 show two access-time traces for the 16-nm technology. The trace with the sharp peak depicts ACS (upper dashed line); the more linear trace just below ACS shows ACI. The lower traces in the plot analytically break ACI down into its several components, such as bank select or precharge time. The large diamonds

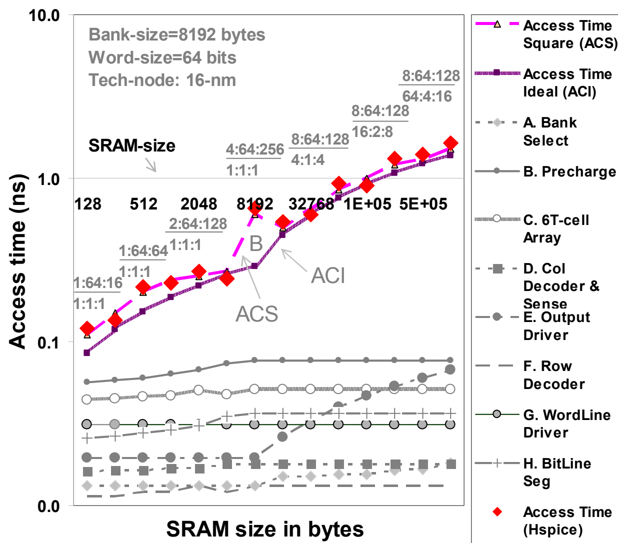


Figure 6: Access-time for "square" SRAM (ACS), Access-time for "non-square" SRAM (ACI), and ACI break-down traces.

surrounding ACS are Hspice results. The number triads listed in Fig.6 (e.g., $\frac{8{:}64{:}128}{16{:}2{:}8}$) represent number of columns(8):word-size(64):number of rows(128) in the upper sets, and total number of banks(16=2×8):number of columns of banks(2):number of rows of banks(8), in the lower sets.

Of the several observations following from Fig.6, we only mention one here. Comparison of the ACS and ACI traces reveals that perfectly square SRAMs do not always produce minimum delays, especially for medium-size units. This finding contradicts previously published work that has found that the minimum delays are always produced by perfectly square SRAMs (and never by non-square SRAMs). However, our model clearly shows that it is possible, in some cases, that the delay of a non-square SRAM can be shorter than the delay of a perfectly square SRAM of the same size. This is possible by selecting an optimum organization and architecture for the SRAM. If one compares the left side of the ACS and ACI traces in Fig.6, it is apparent that SRAM access-time can be reduced up to 31% by favoring one or more SRAM input specifications over others. For example, word-size can be favored over number of rows. This "favoritism" involves only negligible extra area and cost for more sense-amps and flip-flops.

4.4. SRAM yield-estimation model

The D2D and WID variations and, hence, failure probability (P_F) of SRAM is directly related to the yield of the memory chip [13]. To estimate the yield, we use Monte Carlo simulations for D2D distributions of Vth, L, and Vdd (assumed to be Gaussian) in our model. An embedded algorithm takes the result of the Monte Carlo simulation along with the given desired maximum power and area to determine the optimum yield. The algorithm discards the delays not meeting both of the required maximum allowable power and area and selects the smallest delay meeting both the given desired total power and area. For each D2D value of the parameters (say Vth_{D2D}, L_{D2D}, and Vdd_{D2D}) we estimate probability failure (P_F=1-CDF) considering the WID distribution of ΔVth, ΔL, ΔVdd, where CDF is the cumulative distribution function. Finally, the yield is defined as expressed by Mukhopadhyay [13].

$$Yield = 1 - \left(\frac{\sum_{D2D} P_F \left(Vth_{D2D}, Lgate_{D2D}, Vdd_{D2D} \right)}{N_{D2D}} \right) \quad (25)$$

where N_{D2D} is the total number of D2D Monte Carlo simulations (i.e., total number of chips). An increase in the WID variation (i.e., σVth, σL, σVdd) increases the memory-failure probability, thereby reducing the yield.

This means that without proper cell transistor sizing and careful choice of SRAM architecture, yield can suffer significantly. For example, using close to minimum size width for the pull down transistors of each 6T-cell can increase both the read delay and delay variation. Similarly, increasing the number of cells in a column increases capacitance and leakage current of bitlines and also increases the access-time, resulting in increase in P_F, and therefore decline in yield. Hence, for yield enhancement the cell configurations and the memory architecture need to be optimized, considering a given minimum area and power constraints. In this estimation, we have assumed a standard deviation of 4% for D2D distribution of Vth and L, and 2% for Vdd.

Table 2 shows the yield results for 16-nm 64KB SRAM. A similar trend holds for all other sizes of 16-nm SRAM—which is about 3% and 5% lower than the trend observed in 45-nm and 180-nm, respectively. To quantify the approximation error empirically, we compared the results obtained from our model with the empirical results obtained from our actual transistor-level SRAM circuits. The approximation error was below 8%.

Table 2: SRAM yield before and after optimization.

	Architecture	Area	Power	Tac-time	Yield
Initial Design (scaled from 50-nm) [13]	$\frac{8{:}64{:}256}{1{:}1{:}1}$	41 mm²	0.885 W	0.42 ns	57%
Empirically Optimized Designed SRAM	$\frac{64{:}64{:}16}{1{:}1{:}1}$	44 mm²	0.939 W	0.29 ns	93%

5. Conclusion

We present a new method for computing the delay distribution of access-time that considers D2D and architecture-dependent, spatially-correlated WID variations. We propose a model for D2D and WID device threshold voltage, length, and supply voltage variations and show how the delay distribution can be efficiently computed using delay sensitivities. We show how selecting the optimal

architecture can increase the yield in SRAM. Furthermore, we show that perfectly square banks do not necessarily lead to minimum access-times. We significantly extend and enhance the older models by adding both an extra dimension of architectural consideration and additional device parameter fluctuation to their analysis, while producing delay estimates within 4% of Hspice results. The high accuracy of the proposed approach is tested and validated by comparing our results with Monte Carlo simulation and access-time method discussed by Mukhopadhyay and VARIUS [13, 14]. Our model software VAR-TX will be made available online.

6. References

[1] Andrew B. Kahng, Bin Li, Li-Shiuan Peh, and Kambiz Samadi; "ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration," The Design, Automation, and Test in Europe (DATE), '09, pp. 423–428, 2009.

[2] Jaydeep P. Kulkarni, Keejong Kim, Sang Phill Park, and Kaushik Roy; "Process Variation Tolerant SRAM Array for Ultra Low Voltage Applications," Design Automation Conference (DAC), pp. 108-113, 2008.

[3] Steven J.E. Wilton and Norman P. Jouppi; "CACTI: An Enhanced Cache Access and Cycle Time Model," IEEE Journal of Solid-State Circuits (JSSC), vol. 31(5), pp. 677-688, May 1996 (current version: 2002-08-6).

[4] Robert C. Baumann; "Soft Errors in Advanced Semiconductor Devices, Part I: Three Radiation Sources," IEEE Transactions on Device and Materials Reliability (T-DMR), vol. 1(1), pp. 17-22, March 2001.

[5] Luong D. Hung, Masahiro Goshima, and Shuichi Sakai; "SEVA: A Soft-Error- and Variation-Aware Cache Architecture," Pacific Rim International Symposium on Dependable Computing (PRDC) '06, pp. 47-54, 2006.

[6] Baker Mohammad, Martin Saint-Laurent, Paul Bassett, and Jacob Abraham; "Cache Design for Low Power and High Yield," The International Symposium on Quality Electronic Design (ISQED), pp. 103-107, March 2008.

[7] Bharadwaj S. Amrutur and Mark A. Horowitz; "Speed and Power Scaling of SRAM's," International Solid-State Circuits Conference (ISSCC), vol. 35(2), pp. 175-185, February 2000.

[8] Aseem Agarwal, David Blaauw, Vladimir Zolotov, Savithri Sundareswaran, Min Zhao, Kaushik Gala, Rajendran Panda.; "Path-based Statistical Timing Analysis Considering Inter and Intra-die Correlations", In Proceedings of ACM/IEEE International Workshop on Timing Issues (TAU), June 2002.

[9] Xiaoyao Liang and David Brooks; "Latency adaptation of multiported register files to mitigate the impact of process variations", Workshop on Architectural Support for Gigascale Integration (ASGI), 2006.

[10] James W. Tschanz, James T. Kao, Siva G. Narendra, Raj Nair, Dimitri A. Antoniadis, Anantha P. Chandrakasan, and Vivek De; "Adaptive Body Bias for Reducing Impacts of Die-to-die and Within-die Parameter Variations on Microprocessor Frequency and Leakage", IEEE Journal of Solid-State Circuits (JSSC), Vol. 37, pp. 344-539, Nov. 2002.

[11] Shekhar Borkar; "Microarchitecture and Design Challenges for Gigascale Integration", Keynote Speech, 37th International Symposium on Microarchitecture (MICRO), December 2004.

[12] Michael Orshansky, Linda Milor, Pinhong Chen, Kurt Keutzer, and Chenming Hu; "Impact of Spatial Intrachip Gate Length Variability on the Performance of High-Speed Digital Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 21, No. 5, pp. 544-553, May 2002.

[13] Saibal Mukhopadhyay, Hamid Mahmoodi, and Kaushik Roy; "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 24(12), pp. 1859-1880, 2005.

[14] Radu Teodorescu, Brian Greskamp, Jun Nakano, Smruti R. Sarangi, Abhishek Tiwari and Josep orrellas; "VARIUS: A Model of Parameter Variation and Resulting Timing Errors for Microarchitects," IEEE Transactions on Semiconductor Manufacturing (TSM), vol. 21(1), pp. 3-13, 2008.

[15] International Technology Roadmap for Semiconductor (ITRS), http://public.itrs.net, 2011.

[16] ASU Predictive Technology Model (PTM), http://ptm.asu.edu/, 2011.

[17] Puneet Gupta, Andrew B. Kahng, Puneet Sharma, and Dennis Sylvester; "Gate-Length Biasing for Runtime-Leakage Control," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 25(8), pp. 1475-1485, 2006.

[18] David Wolpert, Bo Fu, and Paul Ampadu; "Temperature-Aware Delay Borrowing for Energy-Efficient Low-Voltage Link Design," IEEE International Symposium on Networks-on-Chip (NOCS), pp. 107-114, 2010.

[19] Yun Ye, Frank Liu, Min Chen, Sani Nassif, and Yu Cao; "Statistical Modeling and Simulation of Threshold Variation Under Random Dopant Fluctuations and Line-Edge Roughness," IEEE Transaction on Very Large Scale Integration (VLSI) Systems (TVLSI), vol. 19(6), pp. 987-996, 2011.