

Utah State University

DigitalCommons@USU

---

All Graduate Theses and Dissertations

Graduate Studies

---

12-2012

## Process Variation Aware DRAM (Dynamic Random Access Memory) Design Using Block-Based Adaptive Body Biasing Algorithm

Satyajit Desai  
*Utah State University*

Follow this and additional works at: <https://digitalcommons.usu.edu/etd>

 Part of the [Computer Engineering Commons](#)

---

### Recommended Citation

Desai, Satyajit, "Process Variation Aware DRAM (Dynamic Random Access Memory) Design Using Block-Based Adaptive Body Biasing Algorithm" (2012). *All Graduate Theses and Dissertations*. 1419.

<https://digitalcommons.usu.edu/etd/1419>

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact [digitalcommons@usu.edu](mailto:digitalcommons@usu.edu).



PROCESS VARIATION AWARE DRAM (DYNAMIC RANDOM ACCESS  
MEMORY) DESIGN USING BLOCK-BASED ADAPTIVE BODY BIASING  
ALGORITHM

by

Satyajit Desai

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

of

MASTER OF SCIENCE

in

Computer Engineering

Approved:

---

Dr. Sanghamitra Roy  
Major Professor

---

Dr. Koushik Chakraborty  
Committee Member

---

Dr. Reyhan Bhaktur  
Committee Member

---

Dr. Mark R. McLellan  
Vice President for Research and  
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY  
Logan, Utah

2012

Copyright © Satyajit Desai 2012

All Rights Reserved

## Abstract

Process Variation Aware DRAM (Dynamic Random Access Memory) Design Using  
Block-Based Adaptive Body Biasing Algorithm

by

Satyajit Desai, Master of Science

Utah State University, 2012

Major Professor: Dr. Sanghamitra Roy  
Department: Electrical and Computer Engineering

Large dense structures like DRAMs (Dynamic Random Access Memory) are particularly susceptible to process variation, which can lead to variable latencies in different memory arrays. However, very little work exists on variation studies in DRAMs. This is due to the fact that DRAMs were traditionally placed off-chip and their latency changes due to process variation did not impact the overall processor performance. However, emerging technology trends like three-dimensional integration, use of sophisticated memory controllers, and continued scaling of technology node, substantially reduce DRAM access latency. Hence, future technology nodes will see widespread adoption of embedded DRAMs. This makes process variation a critical upcoming challenge in DRAMs that must be addressed in current and forthcoming technology generations. In this paper, techniques for modeling the effect of random, as well as spatial variation, in large DRAM array structures are presented. Sensitivity-based gate level process variation models combined with statistical timing analysis are used to estimate the impact of process variation on the DRAM performance and leakage power. A simulated annealing-based  $V_{th}$  assignment algorithm using adaptive body biasing is proposed in this thesis to improve the yield of DRAM structures. By applying

the algorithm on a 1GB DRAM array, an average of 14.66% improvement in the DRAM yield is obtained.

(58 pages)

## Public Abstract

Process Variation Aware DRAM (Dynamic Random Access Memory) Design Using  
Block-Based Adaptive Body Biasing Algorithm

by

Satyajit Desai, Master of Science

Utah State University, 2012

Major Professor: Dr. Sanghamitra Roy  
Department: Electrical and Computer Engineering

Process variation can be defined as the deviation of process parameters from its nominal specifications. Variation is induced by several fundamental effects resulting from inaccuracies in the manufacturing equipment. It is a combination of systematic effects (e.g., lithographic lens aberrations) and random effects (e.g., dopant density fluctuations). The effect of process variation becomes particularly important at smaller process nodes, where the variation accounts for a major percentage of nominal length or width of the device. Process variations translate to a wide range in performance metrics of current designs. As technology scales, these die variations are getting larger, significantly affecting performance and compromising circuit reliability. The variation effect of length, width, and oxide thickness variation on the overall delay values of DRAM circuit is evaluated in this thesis. In this work, a novel method to mitigate the effect of process variation on DRAM circuit is proposed. The timing and leakage parameter which determine the performance of the circuit are sensitive to the base voltage of the transistor. A technique which modifies the base voltage of the transistor to try and mitigate the effect of process variation is used in this work. The circuit is first divided into arbitrary blocks. The base voltage of transistors in these blocks are then modified to achieve nominal timing and leakage values for the

DRAM. Simulated annealing-based algorithm is used in order to determine the amount of base voltage required to be applied to each of these blocks.

I would like to dedicate this effort to my parents, my aunt, my grandmother, and my brother.



## Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor, Dr. Sanghamitra Roy, for the continuous support of my master's study and research. Besides my advisor, I would like to thank the rest of my thesis committee for their encouragement, insightful comments, and hard questions.

I thank my fellow labmates at the BRIDGE lab, Saurabh Kothawade, Kshitij Bharamwaj, Dean Ancajas, and Yiding Han, for their help and support in the past two years. Also, I thank my friends Ravi, Tejas, and Neeraj who have provided me with immense support during my two years at Utah State University.

Last but not least, I would like to thank my family and friends.

Satyajit Desai

## Contents

	Page
<b>Abstract</b> . . . . .	<b>iii</b>
<b>Public Abstract</b> . . . . .	<b>v</b>
<b>Acknowledgments</b> . . . . .	<b>viii</b>
<b>List of Tables</b> . . . . .	<b>x</b>
<b>List of Figures</b> . . . . .	<b>xi</b>
<b>Acronyms</b> . . . . .	<b>xiii</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
<b>2 Background and Related Work</b> . . . . .	<b>3</b>
<b>3 DRAM Architectural Model</b> . . . . .	<b>6</b>
<b>4 Process Variation Model</b> . . . . .	<b>13</b>
4.1 Random Variation . . . . .	14
4.2 Systematic Variation . . . . .	14
<b>5 Delay Models</b> . . . . .	<b>19</b>
<b>6 Process Variation Aware DRAM Design</b> . . . . .	<b>25</b>
6.1 ABB Implementation . . . . .	25
6.2 Block-Based $V_{th}$ Assignment Algorithm . . . . .	26
6.2.1 Parametric Yield Function . . . . .	28
6.2.2 $V_{th}$ Assignment . . . . .	28
6.2.3 Type of Moves . . . . .	29
6.2.4 $\gamma_k$ function . . . . .	29
<b>7 Methodology</b> . . . . .	<b>31</b>
<b>8 Validation of the DRAM Model</b> . . . . .	<b>34</b>
<b>9 Results</b> . . . . .	<b>36</b>
<b>References</b> . . . . .	<b>41</b>

## List of Tables

Table	Page
8.1 Validation result. . . . .	35

## List of Figures

Figure	Page
2.1 Variation of delay (normalized to 1) for an CMOS device in an 32nm technology node. . . . .	4
3.1 Basic organization of the DRAM. . . . .	7
3.2 Open bit line DRAM structure. . . . .	8
3.3 Read timing for the asynchronous DRAM. . . . .	9
3.4 Read timing for FPM DRAM. The FPM DRAM holds the row constant for multiple column access in rapid succession. . . . .	10
3.5 Read timing for EDO DRAM. The latch at the output of the EDO DRAM allows for the column access and the data transfer to overlap. . . . .	10
3.6 Read timing for BEDO DRAM. The column access signal is controlled by an internal counter giving it a faster data transfer rate. . . . .	11
3.7 Read timing for a synchronous DRAM. . . . .	12
4.1 Gaussian distribution for parametric variation. . . . .	15
4.2 Systematic variation maps for a die with $\phi = 0.1$ (left) and $\phi = 0.5$ (right). . . . .	17
4.3 Variation map due to systematic and random component (for single systematic error). . . . .	17
4.4 Graph showing the 3D correlation function $\rho(r)$ . . . . .	18
5.1 Delay estimation framework. . . . .	20
5.2 Variation in delay(%) with respect to variation in length(%). . . . .	23
5.3 Variation in delay(%) with respect to variation in width(%). . . . .	23
5.4 Variation in delay(%) with respect to variation in oxide thickness(%). . . . .	24
7.1 Variation in time(%) with respect to variation in $V_{th}$ (%). . . . .	33
7.2 Variation in power(%) with respect to variation in $V_{th}$ (%). . . . .	33

9.1	Delay distribution. . . . .	37
9.2	Delay distribution map in seconds. . . . .	38
9.3	Yield. . . . .	39
9.4	Leakage power. . . . .	39
9.5	Yield in case of lower leakage power limit. . . . .	40
9.6	Leakage power in case of lower leakage power limit. . . . .	40

## Acronyms

CMOS	Complementary Metal Oxide Semiconductor
DRAM	Dynamic Random Access Memory
1T1C	One Transistor and One Capacitor
ECC	Error Correcting Codes
PCM	Phase Change Memory
SRAM	Static Random Access Memory
SDRAM	Synchronous DRAM
FPM	Fast Page Mode
EDO	Extended Data Output
DDR	Double Data Rate
HSPICE	Simulation Program with Integrated Circuit Emphasis
CDF	Cumulative Distribution Function

# Chapter 1

## Introduction

CMOS (Complementary Metal Oxide Semiconductor) scaling has been the catalyst for steady progress in the semiconductor industry for the past three decades. However, steady miniaturization of transistors in the nanometer scale continues to exacerbate the effects of process variation in integrated circuits. These process variations translate to a wide range in performance metrics of current designs. The nature of semiconductor manufacturing process gives rise to both intra-die variations (i.e., device features on one chip can be different) and inter-die variations (i.e., device features across chips can be different). The most severe effects of manufacturing process variation is the uncertainty produced in circuit performance, leakage power, and reliability. These uncertainties substantially complicate circuit design and its optimization; it also and degrades the manufacturing yield of integrated circuits.

Large dense structures like DRAMs (Dynamic Random Access Memory) are particularly susceptible to process variation, which can lead to variable latencies in different memory arrays [1]. Although the design of variation tolerant on-chip SRAM (Static Random Access Memory) caches has received significant attention in the research community [2–7], very little work exists in DRAM variation. This is due to the fact that DRAMs were traditionally placed off-chip and their latency changes due to process variation did not have a significant impact on the overall performance of the system.

However, emerging technology trends like three-dimensional integration (3D stacking), use of sophisticated memory controllers and continued scaling of technology nodes, substantially reduces DRAM access latency [1]. Hence, future technology nodes will see widespread adoption of embedded DRAMs [8]. The main advantages of on-chip ram, also know as embedded DRAMs, are higher memory bandwidth, customized memory sizes, lower power

consumption, and higher system integration. However, embedded DRAM can present considerable challenges in technology and fabrication, performance, testing, design methodologies, and business models [8]. As memory designer look to increase the density of the chip and move to lower technology nodes, designers will face the major roadblock of process variation. This makes process variation a critical upcoming challenge in DRAMs that must be addressed in current and forthcoming technology generations.

In this work, techniques for modeling the effect of random as well as spatial variation in large DRAM array structures are presented. Use of sensitivity-based gate level process variation models combined with statistical timing analysis is made to estimate the impact of process variation on the DRAM performance and leakage power. This thesis also introduces a simulated annealing based  $V_{th}$  assignment algorithm using adaptive body biasing to improve the yield of the DRAM device.

This thesis makes several contributions in the area of robust DRAM design [9].

- This thesis incorporates the effects of random as well as spatial variation in DRAM components (Chapter 4).
- This thesis proposes a sensitivity-based delay model for incorporating the effects of process variation at gate level (Chapter 5).
- This thesis also proposes a  $V_{th}$  assignment algorithm for optimizing the yield of large DRAM arrays (Chapter 6).
- Finally, the results of applying the proposed algorithm on a 1GB DRAM array are reported (Chapter 9).
- On an average, the proposed technique shows a 14.66% improvement in the DRAM yield using the  $V_{th}$  assignment algorithm.



## Chapter 2

### Background and Related Work

Inter die [10] and intra die [11] process variation have significant impact on both timing and power consumption of a chip. Figure 2.1 gives us the amount of variation in delay for an CMOS device in 32nm technology node for variance in length of the device. The figure shows that with slight amount of variation in gate length there is significant impact on delay values of the device. Moreover, the impact of process variation increases as the technology scales down. As a result, future technology nodes will see a severe impact on device performance. Process variation could potentially wipe out advancement provided by one technology generation [12]. The impact of process variation on memory has been analyzed in many previous works [4,13–16]. Work has been done on process variation aware design for both SRAM and DRAM devices.

Agarwal et al. proposed a variation aware SRAM architecture for high performance applications [4]. SRAM cell failures under process variation is analyzed in his work and the proposed architecture adaptively resizes the cache to avoid faulty cells, thereby improving the yield. Process variation tolerant architectures for SRAM have also been discussed by Agarwal et al. [5,6]. A new SRAM cell design to mitigate the effects of variation is proposed by Liang et al. [7]. The proposed architectural change is able to reduce the path stability issue displayed by current generation cell architecture due to process variation. Sasan et al. discussed an variation aware SRAM cache based on voltage frequency scaling [17]. However, the technique requires a cache error map that must be updated whenever there is a change in the operating condition. Mukhopadhyay et al. [18,19] propose a process variation tolerant self repairing SRAM based on body bias technique. Mukhopadhyay et al. [20] propose a statistical sizing methodology to reduce the SRAM cell failures. However, it is a design level technique which ignores the operating condition and the inter-die variation.

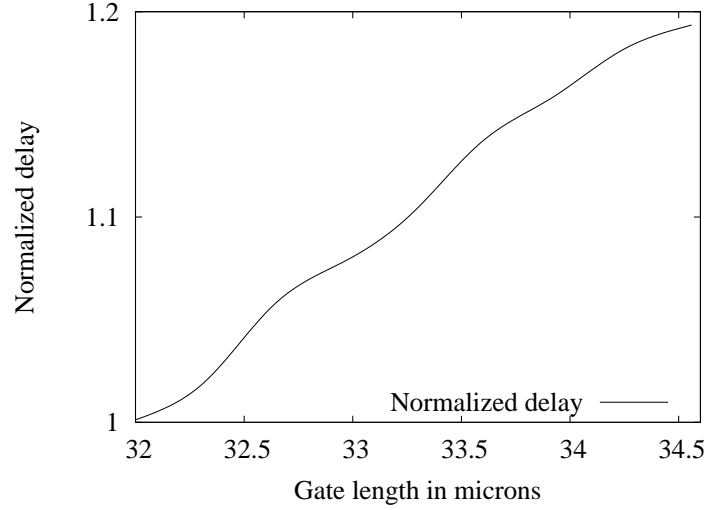


Fig. 2.1: Variation of delay (normalized to 1) for an CMOS device in an 32nm technology node.

Process variation in DRAM has received far less attention in literature as compared to process variation in case of SRAM. Conventionally, errors due to process variation were avoided by providing redundant rows and columns in the design [21, 22]. However, with increasing technology generations, the density of DRAMs is increasing and also the amount of errors due to process variation is on the rise. This type of redundancy technique also has a performance overhead or a resource limitation due to the maximum number of redundant rows and columns that can be included in the design. Thus, use of redundancy-based technique alone will not suffice to avoid process variation in future generation DRAMs. Ohsawa et al. [16] propose a custom hardware support for DRAM chips. The custom hardware is used to refresh different cells at different refresh rates and thereby exploits retention-time variation among memory cells. Kim and Papaefthymiou [15, 23] propose a block-based multiperiod refresh approach, where a custom refresh period is selected for each block. An algorithm to calculate the optimal number of blocks in the system is also provided in their work. Liu et al. [24] also propose a similar approach albeit with a reduction in the area overhead of the system. Venkatesan et al. [25] propose a novel software approach that can exploit off the shelf DRAMs to reduce the refresh power to levels approaching

that of non-volatile memory. When allocating DRAM pages the proposed software prefers longer retention pages as opposed to shorter retention pages which helps to improve power and performance of the device. Mutyam and Narayanan [26] propose the distribution of blocks of cache set over multiple sets, this minimizes the number of sets being affected by process variation. Error correction codes (ECC), which are used for soft error [27,28], can potentially be used for correcting error due to process variation. However, ECC is able to handle only single error, and there is an overhead involved in terms of power consumption, area, and complexity of the design.

All these proposed techniques try to improve the performance of the DRAM device by exploiting the variation in timing parameters or by modifying the behavior of the device through software. The proposed technique is able to reduce the amount of timing variation by use of adaptive base biasing, effectively mitigating the effects of process variation. Thus, resulting in an improvement in performance, yield, and leakage power of the DRAM chip.

In order to mitigate the effect of process variation in DRAM chips, bidirectional base biasing is used. Reverse body biasing techniques have been employed in recent years for reducing the leakage power [29–31]. Reverse body biasing involves lowering the body voltage of a transistor relative to the ground resulting in an improvement in leakage power and a degradation of performance. Forward body bias has also been used in a similar manner to increase the operating frequency of a particular design [32,33]. By combining both these techniques, and controlling the threshold voltages through body bias, the number of dies that meet both frequency and leakage constraints can be maximized. The technique for combining both forward body biasing and reverse body biasing is known as bidirectional body bias [34,35].

## Chapter 3

### DRAM Architectural Model

In this chapter, the basic architectural design of the DRAM model that is used for analysis and the reason for selecting DRAM as a viable option for future memory technology is discussed.

Memory technology, such as PCM (Phase Change Memory) which has superior scalability and power efficiency as compared to DRAM, has also shown considerable promise in case of future memory devices. But, before PCM can be used as an effective alternative to DRAM the significant disadvantages that it introduces as compared to DRAM needs to be eliminated. Writes in case of PCM require a high amount of energy and are slow as compared to DRAM. They require the use of current injection which results in thermal expansion and contraction in the storage cell. Because of this phenomenon PCM has reliability disadvantage restricting the device to hundreds of millions of writes per bit [36,37]. DRAM memory is relatively large, relatively fast, and relatively cheap as compared to other memory technology in the current process node [38]. Moreover, DRAM has been a proven reliable technology which has been employed in modern computer systems since early 1970s. Hence, DRAM is expected to be used in case of embedded memory in near future.

Figure 3.1 gives an overview of the simple architectural structure used in the proposed model. The row decoder circuit, the column decoder circuit, and the DRAM array are present in this model. The DRAM array consists of 1T1C storage cells.

A pre-decoder circuit is incorporated in the design of the decoding circuitry, resulting in multiple levels of hierarchy. Use of the pre-decoder circuit helps to reduce the overall number of gates required for decoding the entire address space.

The DRAM array consists of 1T1C storage cells. Open bitline DRAM array structure is used in the design. Figure 3.2 shows an abstract layout of the open bitline DRAM array

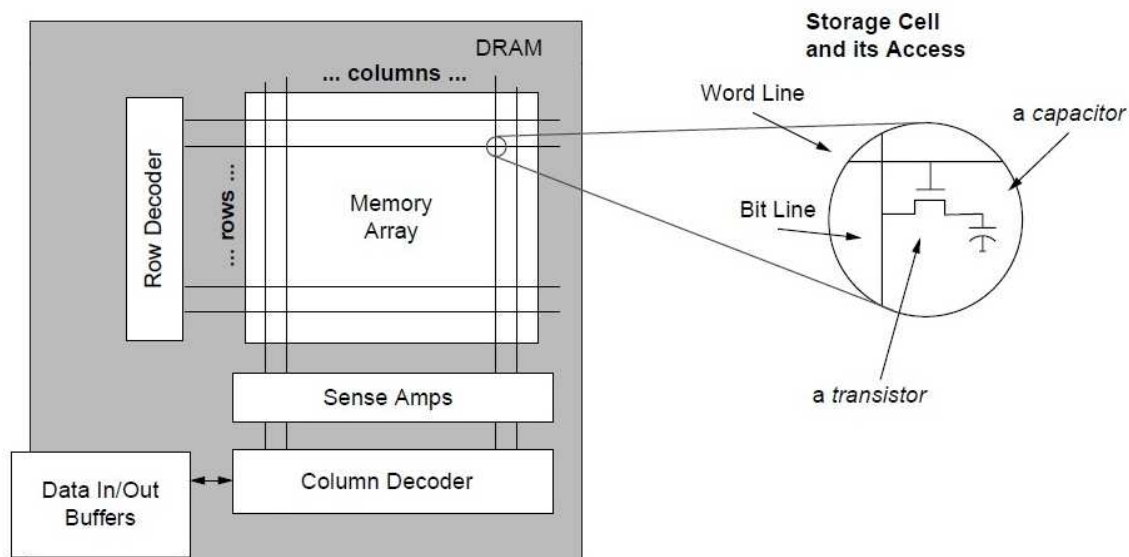


Fig. 3.1: Basic organization of the DRAM.

structure. In the open bitline array structure, bitline pairs used for each sense amplifier come from different array segments [38]. Open bitline array structures are not currently used in modern DRAM devices. However, as process technology advances, open bitline structures promise potentially better scalability for the DRAM cell size in the long run [38].

The core DRAM architecture has remained more or less the same over many technology generations [38]. The DRAM model used in this thesis is abstracted from this core DRAM architecture and the current generation synchronous DRAM. There have been a few hardware modifications to improve the performance of the device. The evolution of these hardware modifications to the core DRAM architecture is discussed in further detail.

Conventional DRAM devices which lead to the development of modern DRAM device were asynchronous in nature. Several improvements were made to the conventional DRAM device which lead to the development of FPM (Fast page DRAM), EDO (extended data out DRAM), and BEDO (burst extended data out device). Although, asynchronous DRAM devices are historical commodity devices and not commercially used, they are important from point of view of understanding the evolution of DRAM devices.

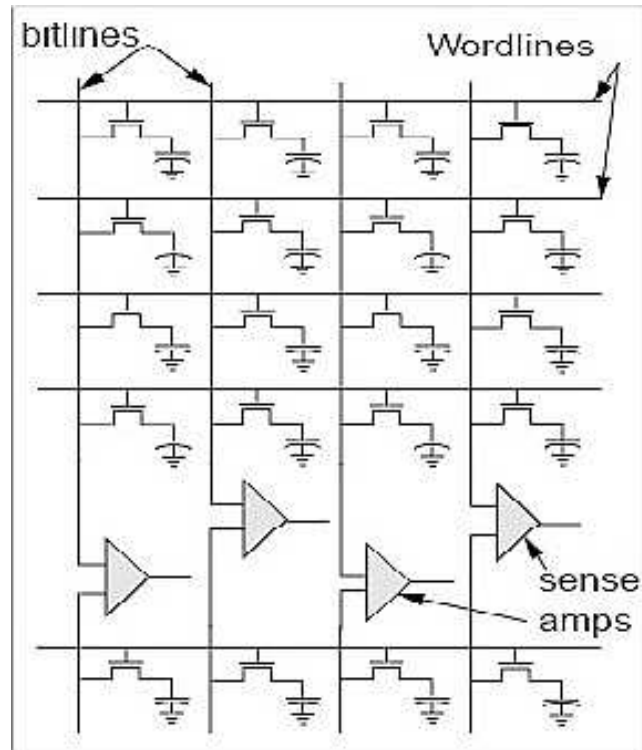


Fig. 3.2: Open bit line DRAM structure.

In case of conventional asynchronous DRAM, the row and column address components are sent separately at two different times on the bus. The row and column address are multiplexed on a single address bus, row access strobe ( $\overline{RAS}$ ) and column access strobe ( $\overline{CAS}$ ) signals are asserted to latch appropriate values at the input. The  $\overline{RAS}$  signal for two sequential accesses to the same row must be de-asserted and re-asserted for a conventional DRAM. Figure 3.3 gives us the timing for a conventional asynchronous DRAM.

In case of most applications it is observed that majority of accesses are to the same row because of spatial locality. This property is exploited in case of a fast-page mode DRAM. The  $\overline{RAS}$  signal may remain asserted in case of fast-page mode DRAM for a faster access cycle as long as the addresses have identical row component addresses. A slight modification is required to the conventional DRAM circuitry to obtain a FPM DRAM [39]. The sense amplifiers for an FPM DRAM have to hold the output value till a change on the address input lines and column address latch takes place. The row address latch holds the same

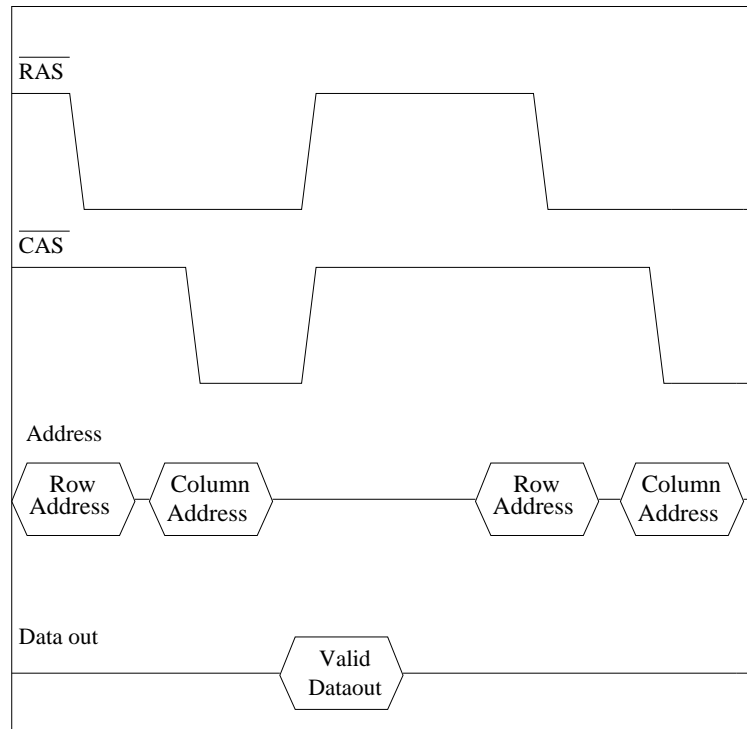


Fig. 3.3: Read timing for the asynchronous DRAM.

value till a change is necessitated by the row address strobe. Figure 3.4 gives the timing for FPM read. There is little increase in the amount of die area for an FPM DRAM as compared to a conventional DRAM.

EDO DRAM followed the fast page DRAM device. The EDO DRAM is a slightly modified version of the FPM DRAM. There is an additional latch present between the sense amplifiers and the output of the DRAM. This helps the DRAM to pre-charge faster, since the output is held by the latch and the  $\overline{CAS}$  can be applied at a higher rate [38]. An BEDO DRAM is an EDO DRAM with a burst counter. Additional control signals are used to differentiate between a normal access and a burst access. Figures 3.5 and 3.6 give us the timing for an EDO read and an BEDO DRAM read, respectively. The amount of die area utilized by these additional modifications is not that significant.

Synchronous DRAM devices was the next major step in the evolution of DRAM devices. The synchronous DRAM device forms the basis of many DRAM devices that followed

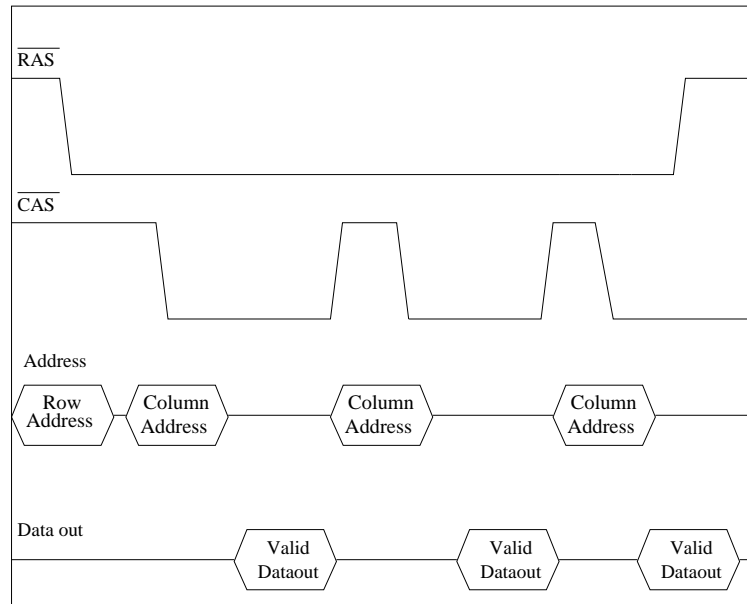


Fig. 3.4: Read timing for FPM DRAM. The FPM DRAM holds the row constant for multiple column access in rapid succession.

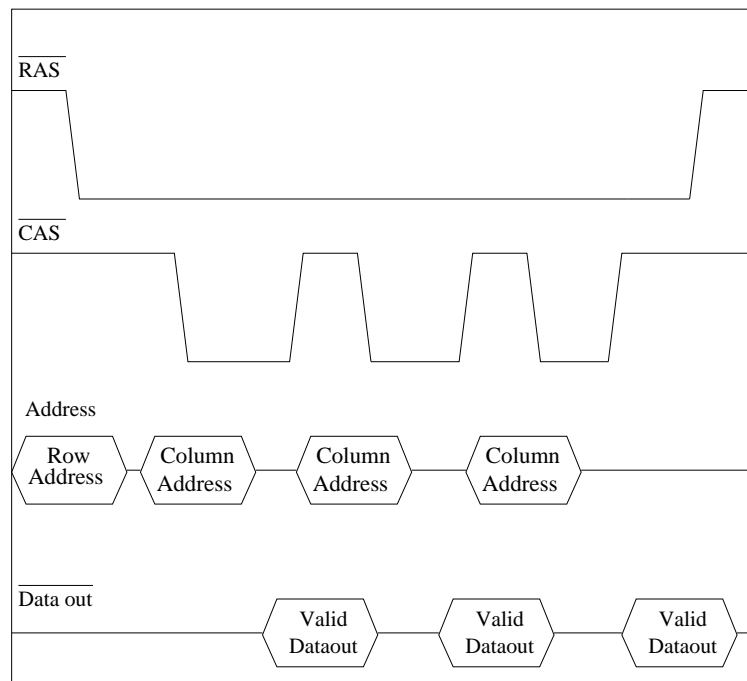


Fig. 3.5: Read timing for EDO DRAM. The latch at the output of the EDO DRAM allows for the column access and the data transfer to overlap.



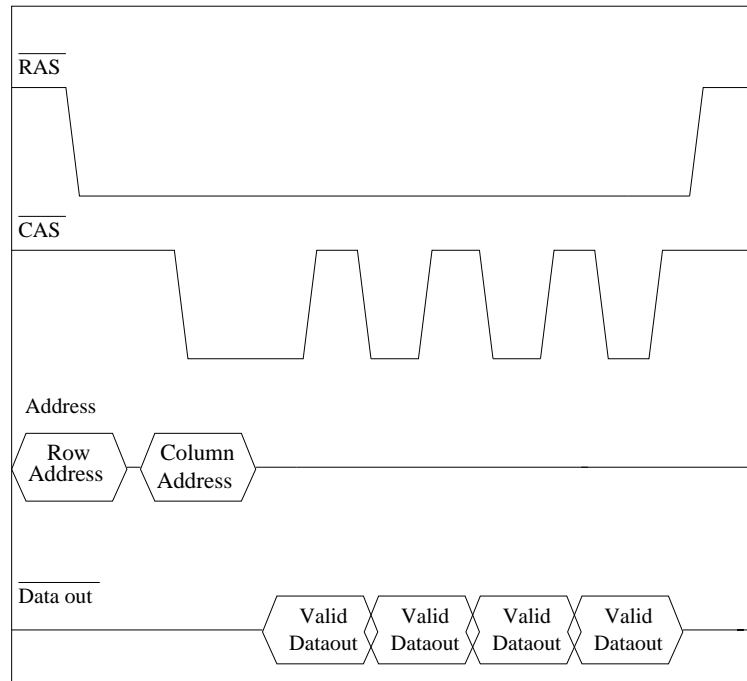


Fig. 3.6: Read timing for BEDO DRAM. The column access signal is controlled by an internal counter giving it a faster data transfer rate.

it. The synchronous DRAM differs from the previous devices in three major ways. The synchronous DRAM device has a synchronous device interface, it contains multiple banks and the synchronous DRAM is a programmable device. The  $\overline{RAS}$  and  $\overline{CAS}$  signal on the synchronous DRAM no longer control the latch directly. A control logic is present in order to evaluate these strobe signals. Figure 3.7 gives us the read timing for an synchronous DRAM. The concept of burst mode is also present in the synchronous DRAM. The Double data rate (DDR) synchronous DRAM improves upon the synchronous DRAM device by operating the data bus at twice the rate of address and command bus [38]. The memory sub system in case of DDR synchronous DRAM transfers data on both the edges of the data strobe signal. Further improvement to the DDR synchronous DRAM is made possible by increasing the prefetch length of the device.

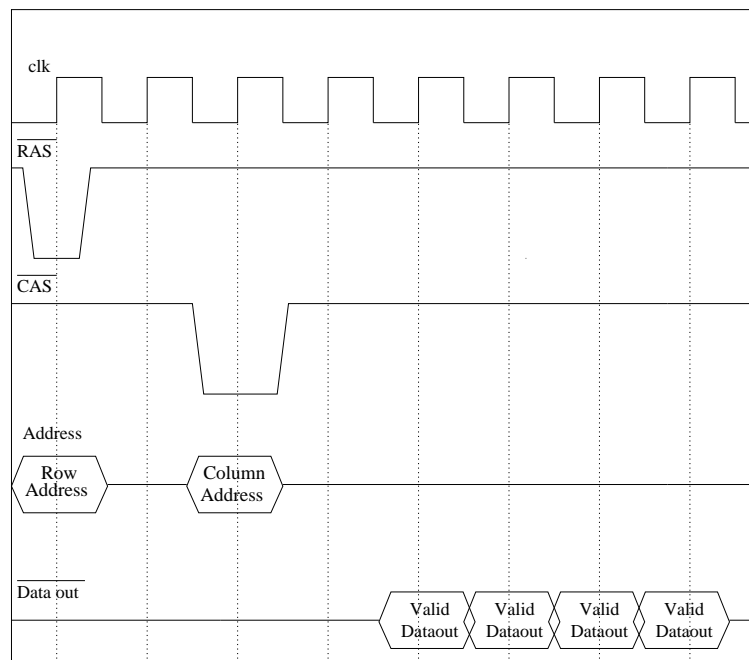


Fig. 3.7: Read timing for a synchronous DRAM.

## Chapter 4

### Process Variation Model

In this chapter, the different models that have been used to account for process variation in DRAMs are discussed.

Process variation is divided into two main categories: inter-die and intra-die variations. Inter-die variation refers to the parametric variation that has a single value across the entire die. Inter-die variation represents a shift in the mean value of the parameter distribution from the nominal value. The inter-die variation parameter captures the variation that occurs from die-to-die, wafer-to-wafer, and from lot-to-lot. These variations are independent from each other, and hence can be represented by a single value for each die. These variations include gate length variations caused from varying exposure time during the manufacturing process and metal thickness variations for different metal layers. There is a systematic trend for inter-die variation across dies that can be captured if the orientation and the location of the die is known during design time. Since, the designer has no control on the placement of the die on the wafer and the information is not available during design time the impact of these factors on the process is captured using random variables. The inter-die variation parameter is assumed to have a simple distribution [40], and hence a Gaussian distribution is used to model it.

Intra-die variation is the component of variation that causes the device parameters to vary across different locations within a single die. Each device on the die is required to have a separate random variable to be able to account for its intra-die variation. Intra-die variations are either spatially correlated or spatially uncorrelated. By definition, systematic variations exhibit spatial correlation and therefore, nearby transistors share similar parametric variation [41]. In contrast, random variation has no spatial correlation, and a transistor's randomly varying parameters differ from those of its immediate neighbors.

Lithographic aberrations introduce systematic variations, while dopant fluctuations and line edge roughness generate random variations.

Most generally, variation in any parameter can be represented as follows [40]:

$$\begin{aligned}\Delta P &= P_{nom} + \Delta P_{inter} + \Delta P_{intra}(X_i, Y_i) \\ &= P_{nom} + \Delta P_{inter} + \Delta P_{spatial}(X_i, Y_i) + \Delta P_{rand,i},\end{aligned}\tag{4.1}$$

where the process parameter that is being affected by variation is represented as  $\Delta P$ .  $P_{nom}$  is the nominal value of the process parameter at a particular technology node.  $\Delta P_{inter}$  is the inter-die variation value and it is constant here as the entire DRAM circuit is assumed to be fabricated from the same die. A Gaussian distribution similar to the one shown in Figure 4.1 is used to model the inter-die variation parameter.  $\Delta P_{intra}(X_i, Y_i)$  is the intra-die process variation affecting a particular block, gate or region “ $i$ ” located at co-ordinates  $X_i$  and  $Y_i$ . This component is further subdivided into spatial ( $\Delta P_{spatial}(X_i, Y_i)$ ) and random component ( $\Delta P_{rand,i}$ ).

#### 4.1 Random Variation

Random variation for a particular parameter is modeled using a Gaussian function. A standard normal distribution is obtained from the Gaussian function by use of Box-Muller transformation. Random variation of different components like length, width, and oxide thickness is mapped using the Gaussian curve shown in Figure 4.1. Random variation occurs at a much finer granularity as compared to systematic variation (i.e., it occurs at the individual transistor level).

#### 4.2 Systematic Variation

The systematic variation or spatial variation is modeled using a normal distribution [38], which has a spherical spatial correlation. Each gate in the row decoder and the column decoder circuit have their own systematic variation parameter. In case of the array

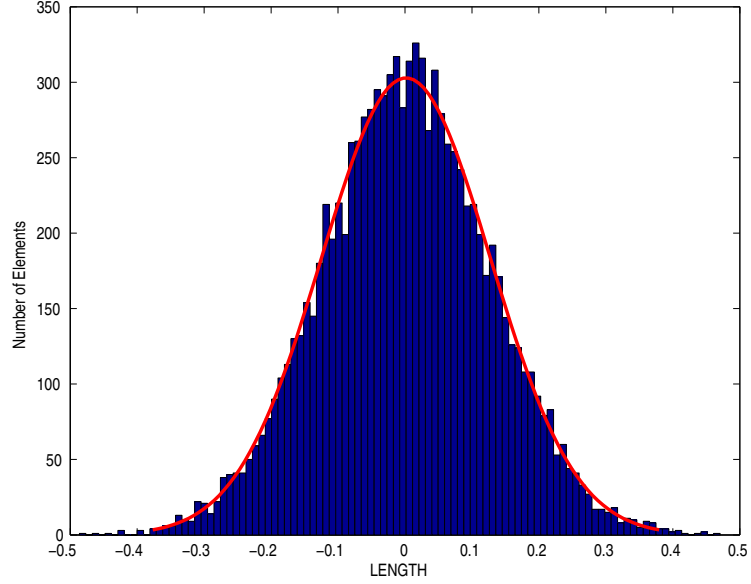


Fig. 4.1: Gaussian distribution for parametric variation.

structure, 16 bits of data line are grouped together as a single block for systematic variation modeling.

Spatial correlation is assumed to be independent of the position on the die and not dependent on the direction. This assumption implies that for any two points on the die, the correlation of systematic variation will be dependent only on the distance between the two points [41]. The correlation function for the systematically varying parameter  $P$  can be defined as

$$\text{Corr}(P_{\vec{x}}, P_{\vec{y}}) = \rho(r) \quad r = |\vec{x} - \vec{y}|, \quad (4.2)$$

where the parameter “ $r$ ” represents the distance between any two devices, or blocks on the die. By definition it can be observed that  $\rho(0) = 1$  and  $\rho(\infty) = 0$ . In order to map the behavior of  $\rho(r)$ , a spherical model is used [41, 42]. The spherical function can be defined as

$$\rho(r) = \begin{cases} 1 - \frac{3r}{2\phi} + \frac{r^3}{2\phi^3} & (r \leq \phi) \\ 0 & \textit{otherwise.} \end{cases} \quad (4.3)$$

The parameter values are highly correlated with their neighbor. The correlation decreases approximately linearly at the beginning. Then it starts decreasing slowly as the value moves further away from the defect point. The spherical model ensures a valid spatial correlation function as defined in Xiong et al. [43].

At a finite distance from the origin of error, the correlation function converges to zero. This means that at a certain distance there is no longer any correlation between the intra-die variation of the two transistors. The limit is defined to be equal to  $\phi$ . In Figure 4.2,  $\phi$  is expressed as a fraction of the chip length. A large  $\phi$  implies that large sections of the chip are correlated with each other; the opposite is true for a small  $\phi$ . As an illustration, Figure 4.2 shows example systematic variation maps for chips with  $\phi = 0.1$  and  $\phi = 0.5$  [41]. In the second case, large spatial features are used, whereas in the first one, the features are small. A distribution without any correlation appears as white noise [41].

A single defect has been introduced on the chip in Figure 4.3, which results in systematic variation. The red spot on the chip represents the area of the chip with maximum deviation from the nominal value due to process variation. The remaining blue area is affected by random variation and the inter-die variation. Thousands of such errors occur during the manufacturing process.

A graphical representation of the above systematic variation function defined in equation (4.3) can be seen in Figure 4.4. The height of the graph corresponds to the amount of variation that is being introduced due to a defect occurring at a particular point on the die. The peak of the 3D graph represents the point where a defect has occurred. From the graph it can be observed that the variation spreads out in a radial pattern from the defect point.

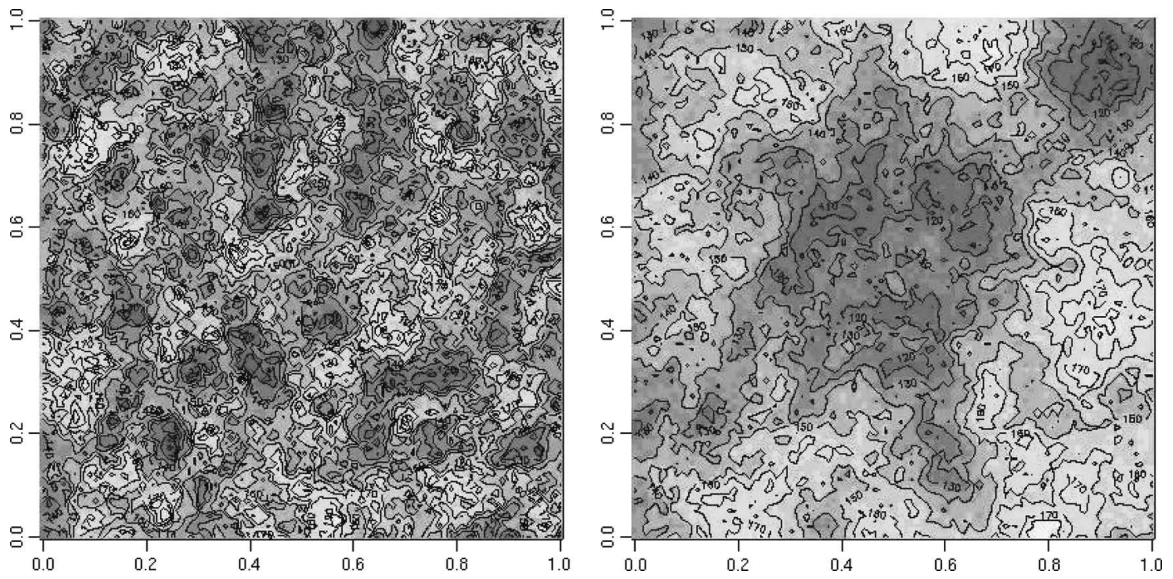


Fig. 4.2: Systematic variation maps for a die with  $\phi = 0.1$  (left) and  $\phi = 0.5$  (right).

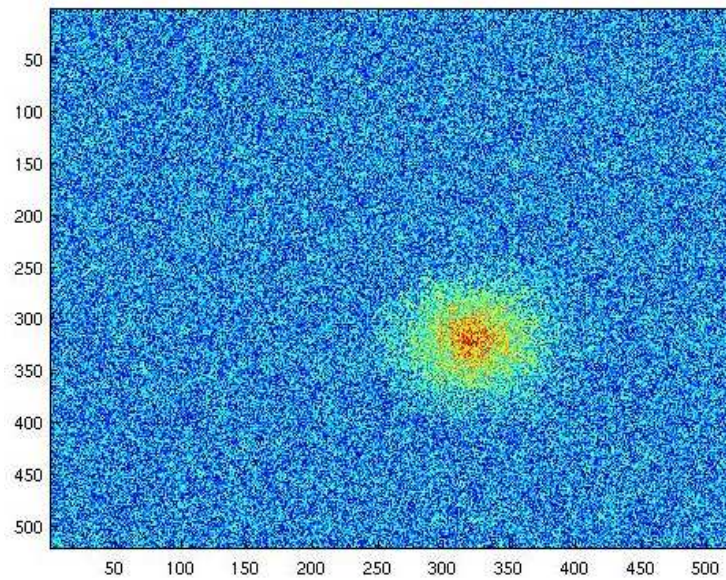


Fig. 4.3: Variation map due to systematic and random component (for single systematic error).

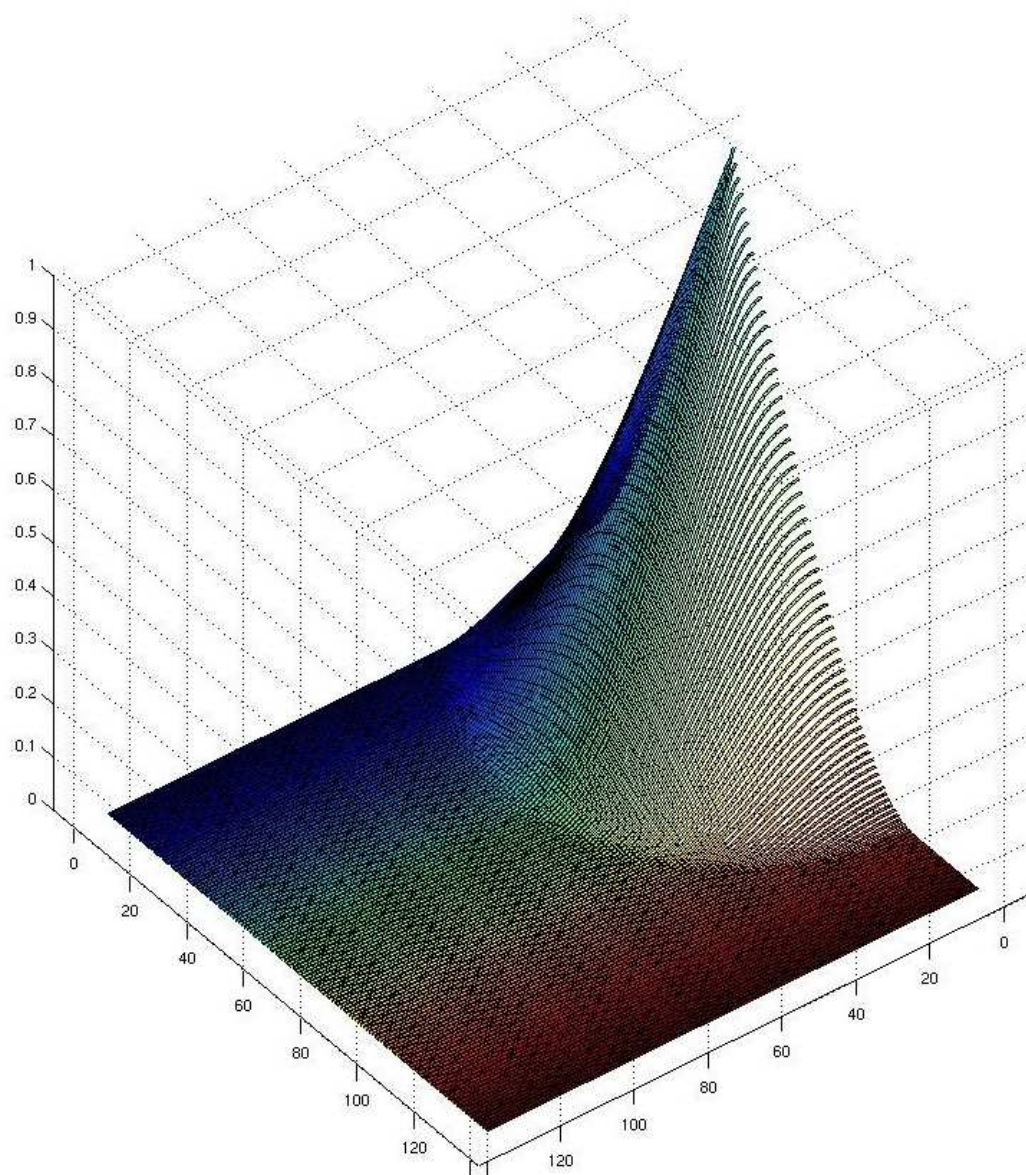


Fig. 4.4: Graph showing the 3D correlation function  $\rho(r)$ .



## Chapter 5

### Delay Models

In this chapter, sensitivity-based delay models for measuring the performance impact of process variation on large DRAM arrays are discussed.

A complete HSPICE (Simulation Program with Integrated Circuit Emphasis) based Monte Carlo simulation for measuring the effect of process variation is computationally prohibitive for large dense structures like a multi GB DRAM. To mitigate this computational effort, a two-step hierarchical modeling approach is taken. HSPICE models are developed for the basic components of the DRAM array including NAND gates, NOT gates, and basic 1T1C DRAM cell. These models are used to develop the look-up tables. The gate level models are used in a statistical timing flow for the DRAM arrays, while incorporating the spatial variation component in the blocks.

Figure 5.1 gives us an overview of the design flow. The intra-die and inter-die variation are combined together using equation (4.1) to form a variation map of the entire DRAM circuit. The variation map gives us the amount of variation present in length, width, and oxide thickness for each device present in the proposed model. Monte Carlo simulations are used to develop look-up table based delay models for NAND, NOT, and the 1T1C memory cell in HSPICE. These look-up tables are employed in the design framework to perform statistical timing analysis of the circuit.

Look-up table-based approach is used for statistical analysis as opposed to analytical equations because the result obtained from HSPICE simulations are not smooth enough to formulate accurate equations. In the proposed approach the transition time at the input and the output loading are used as indexes to find the delay. The delay values for intermediate transition times and output loads are obtained using linear interpolation [40]. The total delay of a particular path is a function of the variation in process parameters

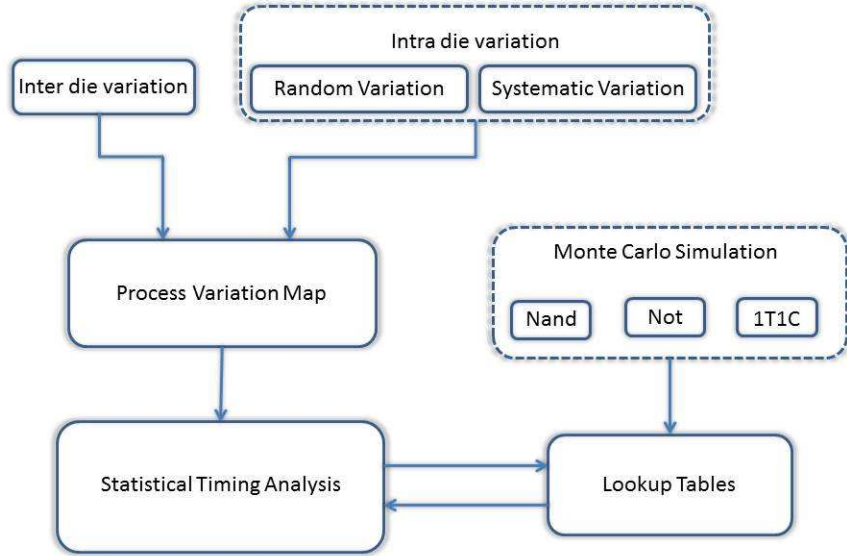


Fig. 5.1: Delay estimation framework.

and is amenable to any arbitrary distribution of the underlying process variation. Linear interpolation is used in order to have a fair amount of accuracy without significant run-time for statistical analysis.

The delay for a particular path is modeled as

$$D_i = D_{nom,i} + \sum_{j=1}^p \sum_{k=1}^p \alpha_{ijk} \Delta P_{jk}, \quad (5.1)$$

where  $D_{nom,i}$  is the nominal value of delay for path “ $i$ .”  $\Delta P_{jk}$  represents the variation in the  $j^{th}$  process parameter of the  $k^{th}$  gate present in path “ $i$ .” The effective variation in gate delay values in response to individual process parameters is captured by the sensitivity parameter  $\alpha$ .  $\alpha_{ijk}$  represents the sensitivity of the parameter “ $j$ ” of gate “ $k$ ” in path “ $i$ ” on the total delay of the path. Three process parameters have been considered (i.e., length, width, and oxide thickness). Figures 5.2, 5.3, and 5.4 present sensitivity analysis for the

variation of delay with respect to length, width, and oxide, respectively. These values are obtained using HSPICE simulations and used to determine the value of the respective sensitivity parameters. The details about the procedure used to obtain these graphs is described in further detail in Chapter 7.

The interconnect network's delay in the open bitline structure is calculated using an RLC circuit model [44]. The sense amplifier delay is assumed to be independent of process variation. A constant delay is assumed for these amplifier circuits.

In order to introduce the interconnect delay in the circuit, RLC interconnect delay model is used [44]. The line resistance, capacitance, and inductance per unit length can be expressed using the equations mentioned by Wong et al. [45] and Qi et al. [46], provided next.

$$R = \rho_{eff} \frac{1}{WT}, \quad (5.2)$$

$$\frac{C_g}{\varepsilon_{ox}} = \left( \frac{2W}{H} \right) + 4.08 \left( \frac{T}{T + 4.53411H} \right)^{0.071} + \left( \frac{S}{S + 0.5355H} \right)^{1.773}, \quad (5.3)$$

$$\frac{C_c}{\varepsilon_{ox}} = 1.4116 \left( \frac{T}{S} \right) \exp \left( \frac{-4S}{S + 8.014H} \right) + 4.08 \left( \frac{T}{T + 4.53411H} \right)^{0.071} + \left( \frac{S}{S + 0.5355H} \right)^{1.773}, \quad (5.4)$$

$$L_{self} = \frac{\mu_0}{2\pi} \left[ \ln \left( \frac{2l}{W + T} \right) + \frac{1}{2} + 0.2235 \left( \frac{W + T}{l} \right) \right], l \gg (W + T), \quad (5.5)$$

$$L_{mut} = \frac{\mu_0}{2\pi} \left[ \ln \left( \frac{2l}{S} \right) - 1 + \frac{S}{l} \right], l > S, \quad (5.6)$$

where  $C_g$  and  $C_c$  are capacitance per unit length between line-to-ground and line-to-line, respectively.  $L_{self}$  and  $L_{mut}$  are self and mutual inductance per unit length, respectively.  $\rho_{eff}$  is the effective resistivity.  $\varepsilon_{ox}$  is the effective dielectric constant of the material and  $\mu_0$  is the vacuum magnetic permeability. W, S, T, H, and l represent the line width, line spacing, line thickness, dielectric thickness, and length of the line, respectively. The interconnect empirical parasitic model that is defined by the formula given above can accurately estimate the electrical parameters for most materials.

The impact of mutual capacitance and inductance on the accuracy of the estimate of interconnect performance is also taken into consideration. The total capacitance and the total inductance per unit length is given as [45]

$$C = C_g + 2C_c,$$

$$L = L_{self} + 2L_{mut}.$$

Combining the above equations for R, L, and C, the value of these parameters can be extracted for the interconnect circuit. An RLC delay model is simulated in HSPICE to obtain the delay value of the interconnect circuit.

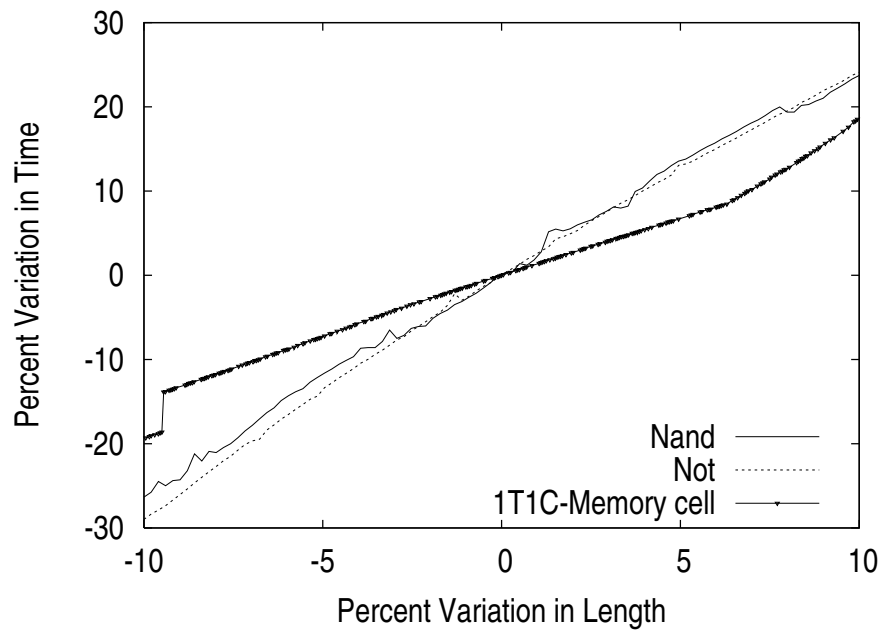


Fig. 5.2: Variation in delay(%) with respect to variation in length(%).

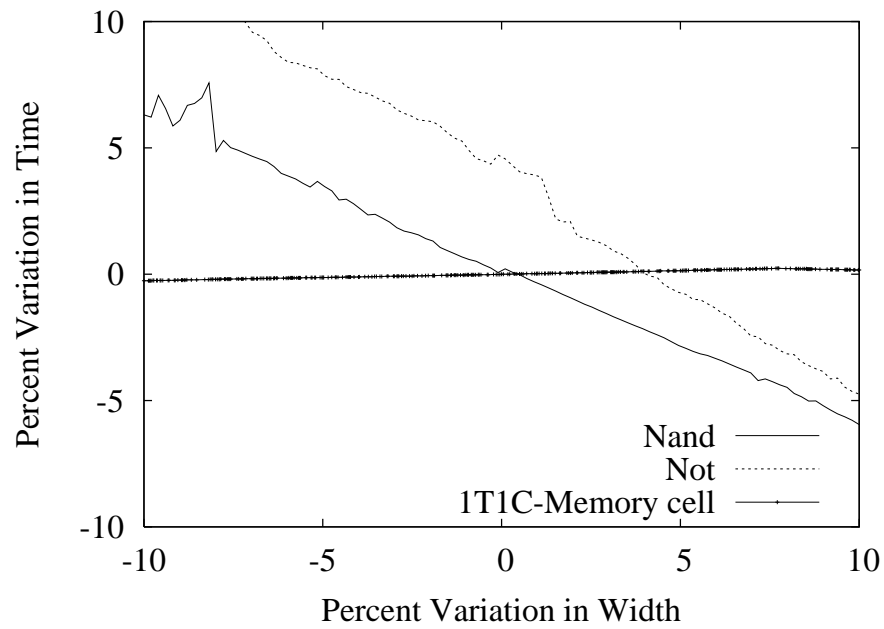


Fig. 5.3: Variation in delay(%) with respect to variation in width(%).

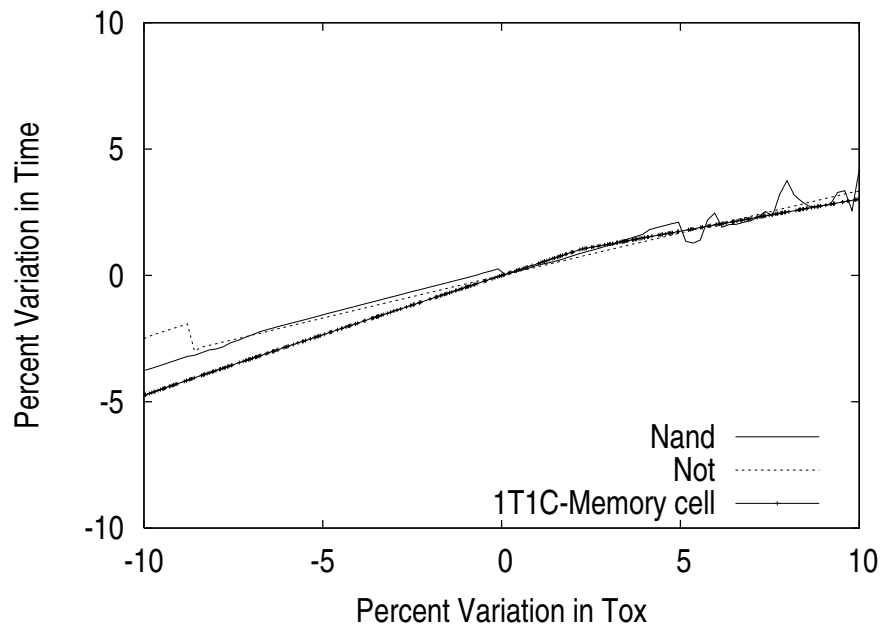


Fig. 5.4: Variation in delay(%) with respect to variation in oxide thickness(%).

## Chapter 6

### Process Variation Aware DRAM Design

In this chapter, technique to make the DRAM model more robust to process variation is developed by use of an adaptive body biasing technique.

Adaptive body biasing is a technique that allows for post-silicon tuning of individually manufactured dies such that each die is able to optimally meet the delay and power constraints. The threshold voltage of the transistor is controlled through body biasing. The delay distribution can either be moved towards the right (by raising the  $V_{th}$ ) or to the left (by lowering the  $V_{th}$ ). The  $V_{th}$  for dies that are too slow is lowered and is increased for dies that are too leaky. This helps to increase the number of dies which meet both the timing and power constraints. Varying the  $V_{th}$  in either direction can be accomplished by the use of bidirectional base biasing [35].

#### 6.1 ABB Implementation

In the simplest ABB scheme, a single bias voltage is used for the entire chip. But the use of a single bias voltage would ignore the intra-die variation. This will result in a sub-optimal solution with a large amount of variation in timing values and leakage among different circuit blocks fabricated on the same chip. In this work, the within die variations are taken into consideration while evaluating the bias voltages. The circuit is divided into individual blocks for the application of bias voltages. The threshold voltage of individual blocks in each die is controlled not only by the manufacturing process but also by the application of appropriate amount of forward base bias and reverse base bias voltages. Each block independently adjusts its own body bias to meet the target delay, and therefore the total die leakage is minimized for the target frequency. Bidirectional ABB is used to apply an optimum threshold voltage, which maximizes the yield subject to timing and

power constraints. The body bias may be applied by an external source or by an on-chip bias generator [35].

A process variation aware method is used that clusters gates at design time into a handful of independent body bias groups or blocks. These base bias groups are then individually tuned post-silicon for each die [47]. The dies are tuned by selecting appropriate  $V_{th}$  voltage for each block. The row decoder circuit, column decoder circuit, and the DRAM array are individually separated into blocks. An appropriate threshold voltage needs to be selected for these independent body bias groups formed during the design stage to achieve maximum yield.

One of the drawback of the adaptive base biasing technique is the need for an additional on-chip distribution network. The distribution network is used for the application of body bias voltages to various blocks in the circuit. The current drawn by these distribution network is small and does not contribute significantly to the overall power consumption of the chip. However, the distribution network occupies additional silicon area and needs to be routed. A central bias generator circuit capable of generating several different bias voltage for each block in the circuit is also required. Thus, in order to limit overhead in the design, the number of blocks to be incorporated in the design should be carefully chosen.

## 6.2 Block-Based $V_{th}$ Assignment Algorithm

Algorithm 6.1 presents the proposed algorithm for robust DRAM design. This algorithm estimates the optimal  $V_{th}$  assignment for each block to maximize the DRAM yield.

There is a large combination of values for  $V_{th}$  assignment of every block. Since, the solution space is very large simulated annealing is used, which is a heuristic algorithm. The delay distribution and power of the circuit is recalculated after every iteration of simulated annealing based on the new  $V_{th}$  assignment. These value are then used to find the yield of the circuit as defined in section 6.2.1. The process of  $V_{th}$  assignment for each block is explained in details in section 6.2.2. There are two moves present in the proposed algorithm described in section 6.2.3.



---

**Algorithm 6.1**


---

```

1: Initialize Model
2: Inflict error in model using spherical model
3: Add Die to Die and Within Die variation
4: Calculate  $r_k$  and  $c_k$  using equations (6.2) and (6.3)
5: Initialize  $V_{th} \leftarrow f(r_k, c_k)$ ; for 'k' number of blocks;
6:  $D \leftarrow DELAY(V_{th})$ ;  $P \leftarrow POWER(V_{th})$ 
7:  $Y \leftarrow YIELD(D, P)$ 
8: while ( $T > \epsilon$ ) OR ( $Reject/moves < 0.90$ ) do
9:    $uphill = moves = Reject = 0$ 
10:  while ( $moves < 2M$ ) OR ( $uphill < M$ ) do
11:     $\gamma_k = f(r_k, c_k, D, P)$ 
12:    Calculate  $V_{th_{new}} \leftarrow RANDOMMOVES(V_{th})$ ; for 'k' number of blocks;
13:     $D_{new} \leftarrow DELAY(V_{th_{new}})$ ;  $P_{new} \leftarrow POWER(V_{th_{new}})$ 
14:     $Y_{new} \leftarrow YIELD(D_{new}, P_{new})$ 
15:     $\Delta Y = Y_{new} - Y$ 
16:    if ( $\Delta Y > 0$ ) OR ( $Random < e^{\Delta Y/T}$ ) then
17:      if ( $\Delta Y > 0$ ) then
18:         $uphill = uphill + 1$ ;
19:      end if
20:       $D \leftarrow D_{new}$ ;  $P \leftarrow P_{new}$ ;
21:       $V_{th} \leftarrow V_{th_{new}}$ ;  $Y \leftarrow Y_{new}$ 
22:      if  $Y > Y_{best}$  then
23:         $Y_{best} \leftarrow Y_{new}$ ;  $V_{th_{best}} \leftarrow V_{th_{new}}$ 
24:      end if
25:    else
26:       $Reject = Reject + 1$ ;
27:    end if
28:  end while
29:   $T \leftarrow \lambda T$ 
30: end while

```

---

### 6.2.1 Parametric Yield Function

The objective function is the parametric yield function, which can be defined as

$$Y = P_r(\text{Delay} \leq D_0, \text{Power} \leq P_0), \quad (6.1)$$

where the delay of the circuit is constrained to be less than  $D_0$  and the leakage power of the circuit is constrained to be less than  $P_0$ .

### 6.2.2 $V_{th}$ Assignment

The DRAM structure in the proposed model is repetitive in nature. This inherent property of the DRAM can be exploited for effective  $V_{th}$  assignment. By use of equation (6.2), and (6.3), approximate value for the spatial correlation of process variation between neighboring devices for a particular block can be found. Equation (6.2) and (6.3) are evaluated for  $n$  number of rows and  $m$  number of columns. The initial  $V_{th}$  assignment is done by using these two equations. Combining these equations with the overall delay and power of the circuit gives us a guideline for future  $V_{th}$  assignment. A parameter  $\gamma$  is created, which is dependent on the overall delay, power,  $r_k$ , and  $c_k$  for this purpose. The  $\gamma$  parameter is evaluated for each block in the circuit and forms the framework for  $V_{th}$  assignment in the next iteration of simulated annealing.

$$r_k = \sum_{j=0}^m \frac{t_{i,j}}{m} \text{ for } i = 0 \text{ to } i = n - 1, \quad (6.2)$$

$$c_k = \sum_{i=0}^n \frac{t_{i,j}}{n} \text{ for } j = 0 \text{ to } j = m - 1, \quad (6.3)$$

where  $t_{i,j}$  is the time required to access the memory cell located in the " $i^{th}$ " row and " $j^{th}$ " column of the DRAM array. " $k$ " denotes the different blocks present in the circuit.

### 6.2.3 Type of Moves

In this algorithm, two moves,  $m1$  and  $m2$  defined in (6.4) and (6.5), are used. One of the moves is randomly selected for computing the new  $V_{th}$  assignment of the “k” number of blocks present in the circuit. In the first move  $m1$ , previously calculated function “ $\gamma$ ” is used, which is dependent on timing delay, leakage power, and the spatial correlation of error in the circuit. For the second move  $m2$ , the  $V_{th}$  level of each circuit block are randomly varied.

$$m1 : V_{th}(block_k) = V_{th}(block_k) + \gamma_k \times V_{th}(block_k) \quad (6.4)$$

$$m2 : V_{th}(block_k) = V_{th}(block_k) \pm rand \times V_{th}(block_k) \quad (6.5)$$

The delay distribution and the leakage power of the circuit is then recalculated based on the new  $V_{th}$  assignment.

### 6.2.4 $\gamma_k$ function

The value of the  $\gamma$  function is based on the leakage power, timing distribution, and spatial correlation of individual blocks. Each block in the model has a unique value of  $\gamma$  associated with it.

Decrease in the  $V_{th}$  level for a block would result in reduction of leakage power and an increase in timing distribution [29], whereas an increase in the  $V_{th}$  level would result in improvement to the timing distribution of the block and increase in power [32]. According to equation (6.4), a negative value for the  $\gamma$  function is required to improve the leakage power. On the other hand, a positive value for the  $\gamma$  function would help to improve the timing distribution of the block. The magnitude of the  $\gamma$  function is proportional to the deviation of leakage power and the timing delay from its nominal values for a particular block.

Lower value of deviation implies that the delay values might be related strongly to each other. Thus, the effect that the  $\gamma$  function has on the overall block is consistent. A smaller magnitude of  $\gamma$  would suffice to meet the timing and power requirement. On the other hand,

if the deviation among delay values in a block is high, transistor lying at extremities are to be considered. To handle these large disparities between the transistors dimensions, the worst case condition is taken into account. So, a higher value for the  $\gamma$  function is reached in this case.

The size of the block is also an important consideration for determining the value of  $\gamma$ . A larger block size would result in a rapid increase in the overall leakage power due to increase in the  $V_{th}$  levels. This usually results in a failure for the chip as a result of the power constraints. Thus, a more conservative policy is employed for determining the value for the  $\gamma$  function in case of large blocks. In case of smaller blocks, a much more aggressive policy can be employed.

## Chapter 7

### Methodology

In this chapter, the various HSPICE models that have been used for delay calculations are discussed. The technique used to calculate the new delay and leakage power values after the new  $V_{th}$  assignment is also discussed in this chapter.

Monte Carlo simulations are performed for NAND, NOT, and 1T1C memory storage cell in HSPICE to obtain the variation in delay values with changes in process parameters (i.e., with variation in length, width, and oxide thickness of the transistor). Values of length, width, and oxide thickness were varied with a uniform distribution, with variance of 0.1 [48]. The data obtained from these simulations is shown in Figures 5.2, 5.3, and 5.4. The corresponding values that were obtained are used to form the look-up tables. The access timing of the DRAM circuit was obtained from these look-up tables using equation (5.1).

The leakage power is calculated using the leakage power model provided by Srivastava et al. [49]. The leakage power model can be expressed as

$$Leakage = exp(V_{nom} + \sum_{i=1}^p \beta_i \Delta P_i), \quad (7.1)$$

where  $exp(V_{nom})$  is the nominal value of leakage power and  $\Delta P_i$  represents the variation in the process parameter. The sensitivity of the process parameter to the overall leakage power of the device is expressed in terms of  $\beta$ . The total leakage power is expressed as the sum of the leakage power of individual gates amenable to the underlying variation. A similar approach to the delay calculations is taken for the leakage power calculations. A look-up table approach is used in order to obtain the leakage power of the different circuit element present in the device based on the variation in process parameters. The look-up tables are obtained by performing Monte Carlo simulations for the various devices in the

circuit to determine the leakage power variations.

In order to compute the new yield using the simulated annealing algorithm, the delay and leakage power values of the DRAM circuit are recalculated based on the  $V_{th}$  assignment for the various blocks present in the circuit. The static leakage power is taken into account to estimate the yield, because variations in dynamic power due to process variation are insignificant as compared to static leakage power [49]. To calculate these new delay values, equation (5.1), equation (7.1), and the look-up tables obtained from the graphs in Figures 7.1 and 7.2 are used. The sensitivity parameter for HSPICE simulations is dependent only on the change in  $V_{th}$  levels. Based on the percentage change in the  $V_{th}$  value present in the new voltage assignment, the new delay and power values are determined.

Two chips are simulated, each with four 128MB banks. There is one chip select bit and two bank select bits. A total of 13 row address bits and 11 column address bits are used in order to access data in these banks. The two chips are assumed to be from the same die. In this case, the 32nm technology node is used for simulation purpose and this is the nominal value. Ten-thousand Monte Carlo simulations are performed to obtain the final delay values for all the DRAM bits present in the array. All the column bits present are accessed in a single row access in order to obtain the random access time  $t_{RAC}$ .

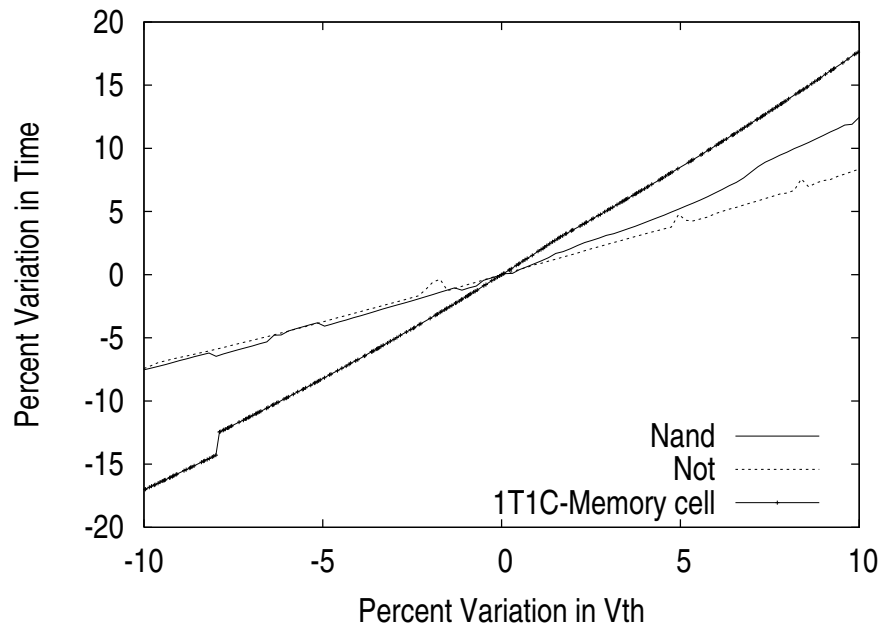


Fig. 7.1: Variation in time(%) with respect to variation in  $V_{th}$ (%).

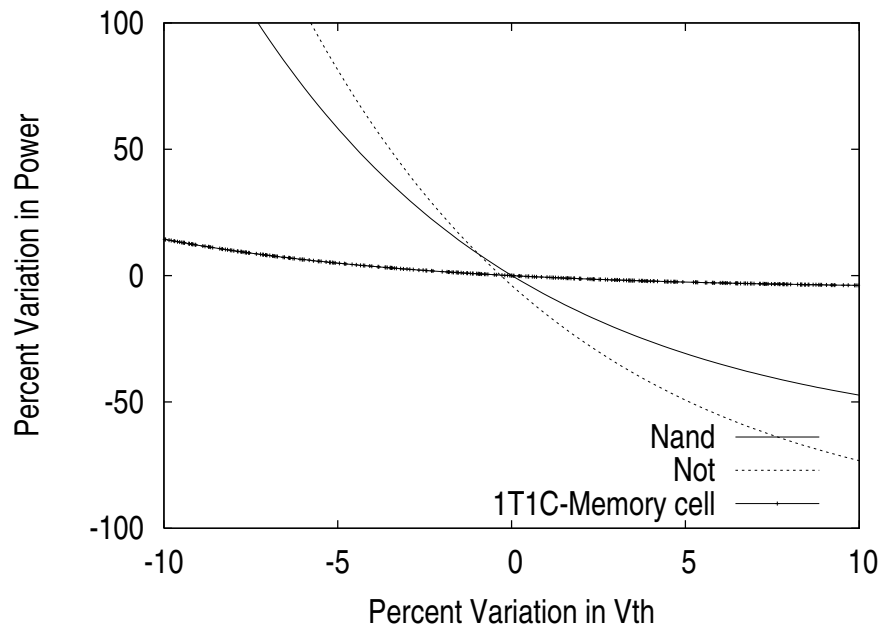


Fig. 7.2: Variation in power(%) with respect to variation in  $V_{th}$ (%).

## Chapter 8

### Validation of the DRAM Model

To verify the result obtained for the delay model described in Chapter 5, complete Monte Carlo simulation is used. The validation of the proposed model has been done through HSPICE simulations for a 32nm CMOS process. Sixteen-bit, 32-bit, and 64-bit DRAM modules are modeled in HSPICE and Monte Carlo simulations are performed to obtain a reliable timing distribution. The DRAM module created in HSPICE is based on the architectural DRAM model that is described in Chapter 3. The 16-bit DRAM device is arranged in a 4X4 matrix formation with four row and column address bits to resolve the address space. Likewise, the 32-bit DRAM device is arranged in a 8X4 matrix formation and the 64-bit DRAM device has a 8X8 matrix formation. The row decoder and the column decoder circuit consists of NAND and NOT gates. Row access strobe ( $\overline{RAS}$ ) and column access strobe ( $\overline{CAS}$ ) signals are used in the circuit in order to simulate the access pattern of a conventional DRAM.

In order to validate the model, a reliable timing distribution should be obtained from the HSPICE simulations. The access time for a particular bit in the simulations is calculated as the time beginning from the application of the row address strobe to the time it takes for output to arrive (i.e.,  $t_{RAC}$ ). In order for timing values to be consistent with the access pattern of a standard DRAM device, the row address strobe is applied first to resolve the row address line. The column access strobe is applied only after the row address line is resolved by the decoder circuit. To simulate the effect of process variation in case of HSPICE models, process variation model describe in Chapter 4 is used. The error map obtained from the process variation model gives us the total effect intra-die variation and inter-die variation has on all the gates and DRAM array present in the proposed model. The variation that occurs in case of the process parameters length, width, and oxide thickness for each gate



and storage cell is present in the error map. The equivalent process parameters are modified for gates and storage cell present in the HSPICE model using the error map to simulate the overall effect of process variation. Since, the proposed process variation model is used, inter-die variation, random variation, and systematic variation are all taken into account in case of the HSPICE model. One-thousand Monte Carlo simulations are performed in order to obtain the delay distribution for comparison of mean and standard deviation of the timing access. Table 8.1 gives us the error for mean and standard distribution of the timing values for the various DRAM models.

From the results obtained it can be observed that there is an error present for the mean and the standard deviation of the timing parameter. The look-up table approach taken for out statistical analysis uses linear interpolation to calculate delay values for intermediate transition times. The use of linear interpolation for determining the timing values of the circuit paths adds some amount of error to the simulation result. The error due to linear interpolation could be reduce by use of quadratic interpolation method albeit at the cost of increase in computational complexity of the model, which will result in a huge increase in the run-time.

Table 8.1: Validation result.

DRAM Size	Error Compared to Monte Carlo Simulations	
	Mean Error	Standard Deviation Error
16 Bit DRAM	5.03%	3.77%
32 Bit DRAM	5.18%	2.28%
64 Bit DRAM	4.45%	0.76%

## Chapter 9

### Results

The CDF (Cumulative Distribution Function) plot of the access delay for a 16-bit DRAM array in terms of  $t_{RAC}$  is shown in Figure 9.1.  $t_{RAC}$  is the total time taken by the DRAM from the point it gets the address input, to the time it puts the values on the output pins.  $t_{RAC}$  can also be expressed as the addition of row command delay (i.e.,  $t_{RCD}$  and column access strobe latency,  $t_{CAS}$ ). The CDF plot is used as an example to illustrate the effect process variation has on the overall performance of the device. The CDF plot indicates that some of the bits have a higher timing values as compared to others due to process variation. Some section of the die may have a positive change in timing values while other sections may have a negative impact due to variation.

The distribution map is obtained by plotting the timing values of different bits in the DRAM array. The entire chip is divided into 16 blocks in this case for adaptive base bias application. The distribution map for the time required to access individual storage bits in absence of adaptive base biasing is shown in Figure 9.2(a). The varying effect that process variation can have on different section of the die due to intra-die variation is illustrated in the distribution map. The change in the distribution map after the application of adaptive base biasing can be seen in Figure 9.2(b). The distribution map indicates that there is a reduction in the delay for nearly all the blocks in the circuit. Some of the delay values lying on the corner cannot be reduced further because of leakage power restrictions.

The yield result for adaptive base bias implementation are shown in Figure 9.3 and the equivalent leakage power result are shown in Figure 9.4. The yield result have been obtained for a 1GB DRAM using 2, 4, 8, and 16 blocks. On average, 14.66% improvement is seen in the yield. The yield result shown in Figure 9.5 are obtained by decreasing the leakage power limit in the yield function. The resulting change in leakage power is shown in

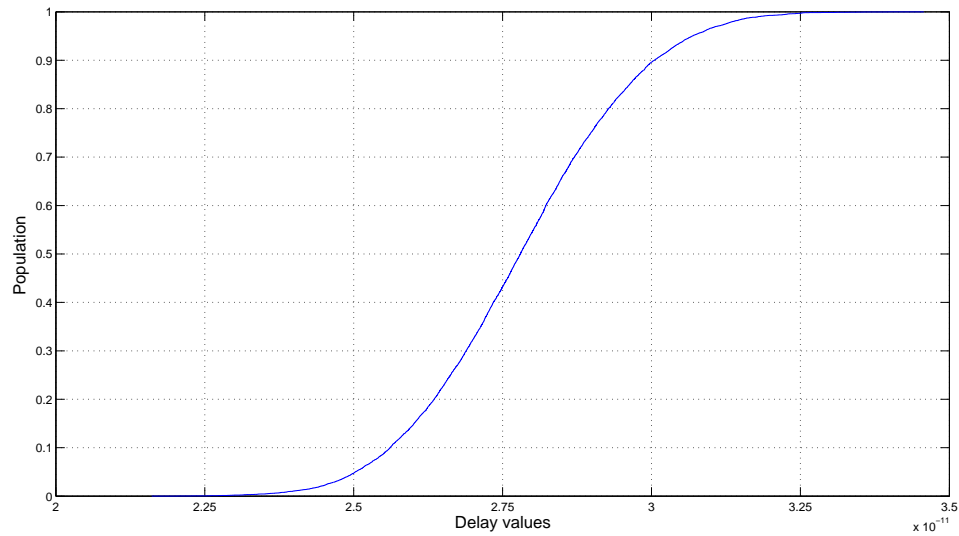
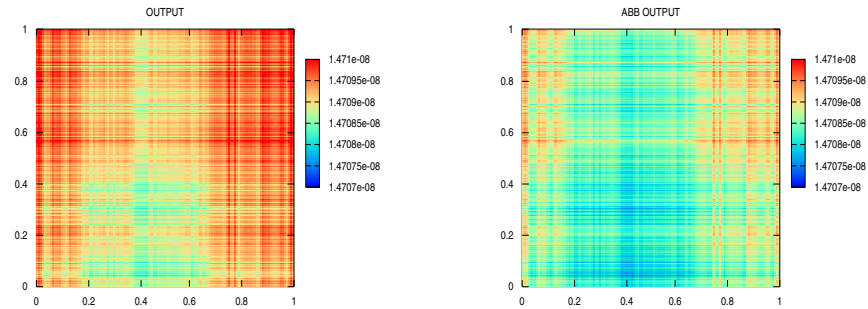


Fig. 9.1: Delay distribution.

Figure 9.6. As a result of a more stringent requirement for power, the overall yield decreases as more dies fail due to the restrictions brought about by the leakage power constraint. An average improvement of 6.07% is seen for yield in this case. The number of blocks in the die is an important parameter that needs to be carefully selected during design phase. There is a tradeoff in terms of the yield of the device and the design complexity, leakage power, and area which needs to be carefully evaluated beforehand.

In both cases, there is an increase in yield when moving from 2 to 16 blocks. This is due to the fact that there is better granularity of control over leakage power and timing values as the number of blocks increases, but there is a trade-off in terms of design complexity and area. The better control translates into better utilization of the timing slack or the power slack and results in an improvement to the overall yield of the device. The leakage power of dies that are too leaky is reduced and improves the timing values of the die which are too slow by use of the proposed technique. The relation between the threshold voltage and increase in leakage power is shown in Figure 7.2. Because of the exponential nature of the relation, the increase in leakage power for slow dies is much higher as compared to the decrease in leakage power achieved in case of leaky dies. The size of the block is also an



(a) Delay map of DRAM array. (b) Delay map of DRAM array after ABB application.

Fig. 9.2: Delay distribution map in seconds.

important factor in determining the effect the proposed technique has on the leakage power. A larger block size results in a rapid increase in the overall leakage power due to increase in the forward base biasing voltage. This usually can result in a failure for the die as a result of the power constraints. A smaller block size on the other hand is able to better manage the leakage power slack. Thus, the technique is able to achieve a better yield by pushing the overall leakage power of the die to the limit. Therefore, there is an increasing trend in leakage power of the device due to the use of adaptive base biasing technique.

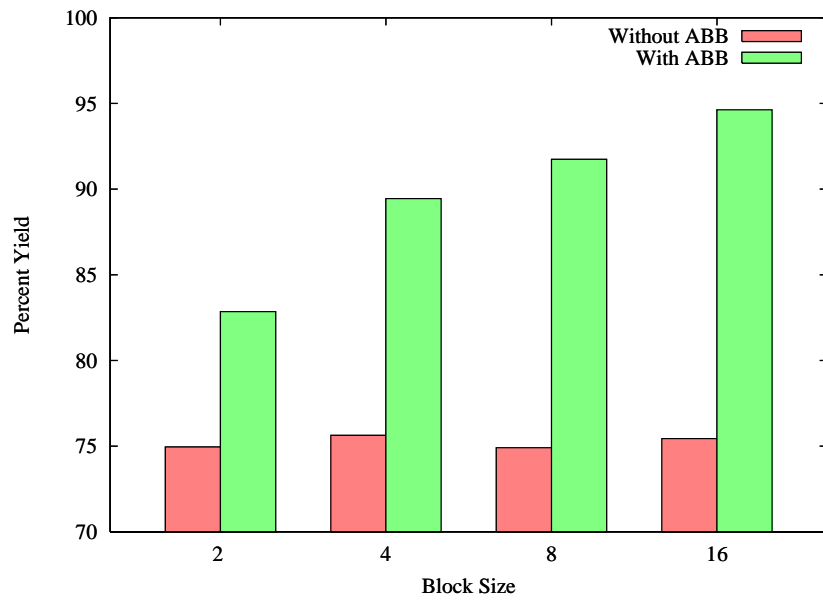


Fig. 9.3: Yield.

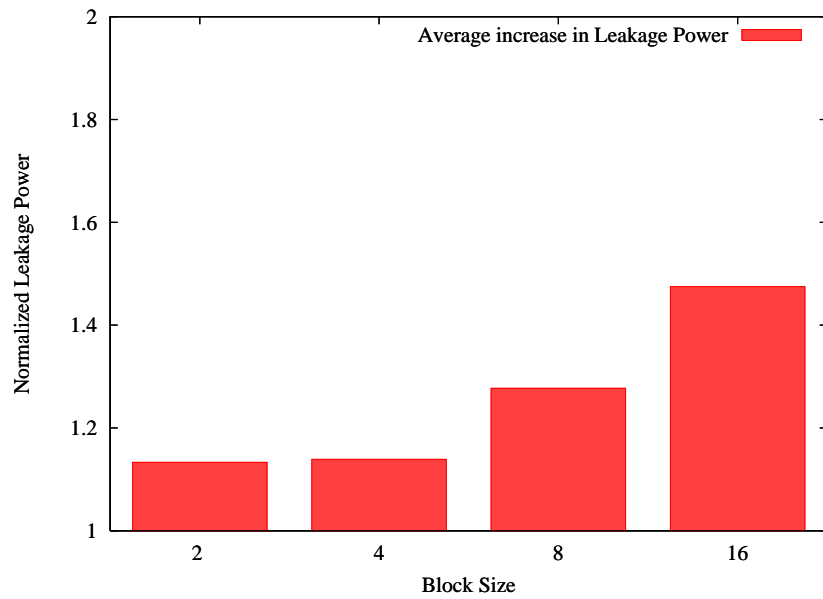


Fig. 9.4: Leakage power.

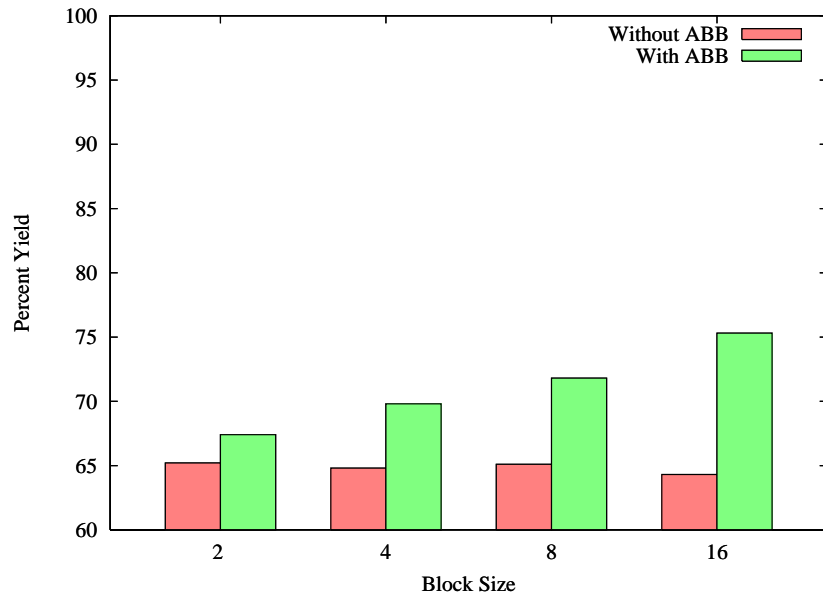


Fig. 9.5: Yield in case of lower leakage power limit.

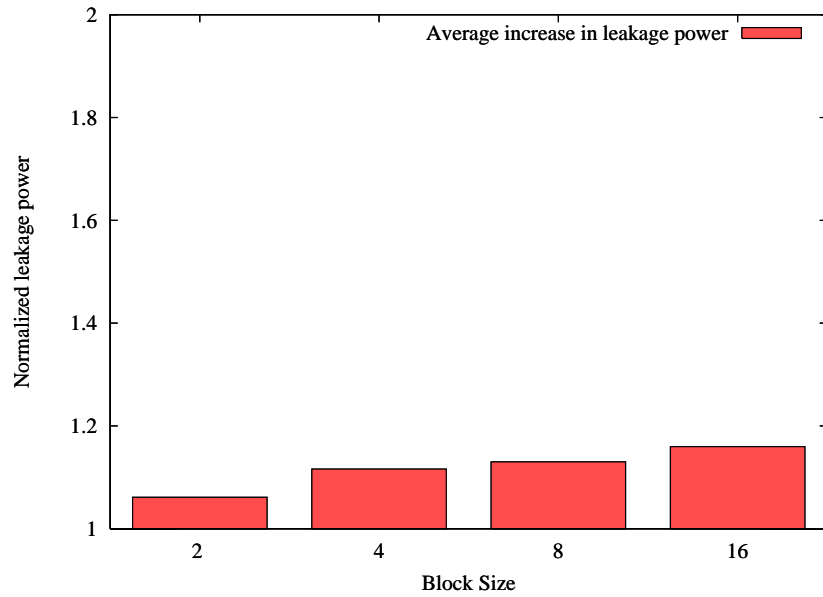


Fig. 9.6: Leakage power in case of lower leakage power limit.

## References

- [1] B. Zhao, Y. Du, Y. Zhang, and J. Yang, "Variation-tolerant non-uniform 3d cache management in die stacked multicore processor," in *MICRO 42: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2009.
- [2] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of sram array for yield enhancement in nanoscaled cmos," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1859–1880, Dec. 2005.
- [3] J. Lee and A. Davoodi, "Comparison of dual-vt configurations of sram cell considering process-induced vt variations," in *ISCAS 2007*, pp. 3018–3021, May 2007.
- [4] A. Agarwal, B. Paul, S. Mukhopadhyay, and K. Roy, "Process variation in embedded memories: failure analysis and variation aware architecture," in *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1804–1814, Sep. 2005.
- [5] A. Agarwal, B. C. Paul, H. Mahmoodi, A. Datta, and K. Roy, "A process-tolerant cache architecture for improved yield in nanoscale technologies," in *IEEE Transactions on Very Large Scale Integration System*, vol. 13, no. 1, pp. 27–38, Jan. 2005.
- [6] A. Agarwal, B. Paul, and K. Roy, "Process variation in nano-scale memories: failure analysis and process tolerant architecture," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 353–356, Oct. 2004.
- [7] X. Liang, R. Canal, G.-Y. Wei, and D. Brooks, "Process variation tolerant 3t1d-based cache architectures," in *40th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 15–26, Dec. 2007.
- [8] D. Keitel-Schulz and N. Wehn, "Embedded dram development: Technology, physical design, and application issues," in *IEEE Design Test of Computers*, vol. 18, no. 3, pp. 7–15, May 2001.
- [9] S. Desai, S. Roy, and K. Chakraborty, "Process variation aware dram design using block based adaptive body biasing algorithm," in *13th International Symposium on Quality Electronic Design (ISQED)*, pp. 255–261, Mar. 2012.
- [10] S. Nassif, "Within-chip variability analysis," in *International Electron Devices Meeting*, pp. 283–286, Dec. 1998.
- [11] —, "Modeling and analysis of manufacturing variations," in *IEEE Conference on Custom Integrated Circuits*, pp. 223–228, 2001.
- [12] K. Bowman, S. Duvall, and J. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," in *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, pp. 183–190, Feb. 2002.

- [13] Q. Chen, H. Mahmoodi, S. Bhunia, and K. Roy, "Modeling and testing of sram for new failure mechanisms due to process variations in nanoscale cmos," in *Proceedings of the 23rd IEEE Symposium on VLSI Test*, pp. 292–297, 2005.
- [14] S. Mukhopadhyay, H. Mahmoodi-Meimand, and K. Roy, "Modeling and estimation of failure probability due to parameter variations in nano-scale srams for yield enhancement," in *Symposium on VLSI Circuits*, pp. 64–67, Jun. 2004.
- [15] J. Kim and M. Papaefthymiou, "Block-based multiperiod dynamic memory design for low data-retention power," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 6, pp. 1006–1018, Dec. 2003.
- [16] T. Ohsawa, K. Kai, and K. Murakami, "Optimizing the dram refresh count for merged dram/logic lsis," in *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 82–87, Aug. 1998.
- [17] A. Sasan (Mohammad A Makhzan), H. Homayoun, A. Eltawil, and F. Kurdahi, "Process variation aware sram/cache for aggressive voltage-frequency scaling," in *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 911–916, 2009.
- [18] S. Mukhopadhyay, K. Kim, H. Mahmoodi, and K. Roy, "Design of a process variation tolerant self-repairing sram for yield enhancement in nanoscaled cmos," in *IEEE Journal of Solid-State Circuits*, vol. 42, no. 6, pp. 1370–1382, Jun. 2007.
- [19] S. Mukhopadhyay, K. Kang, H. Mahmoodi, and K. Roy, "Reliable and self-repairing sram in nano-scale technologies using leakage and delay monitoring," in *Proceedings of IEEE International Test Conference*, pp. 1125–1135, Nov. 2005.
- [20] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Statistical design and optimization of sram cell for yield enhancement," in *IEEE/ACM International Conference on Computer Aided Design*, pp. 10–13, Nov. 2004.
- [21] S. Schuster, "Multiple word/bit line redundancy for semiconductor memories," in *IEEE Journal of Solid-State Circuits*, vol. 13, no. 5, pp. 698–703, Oct. 1978.
- [22] M. Horiguchi, "Redundancy techniques for high-density drams," in *Second Annual IEEE International Conference on Innovative Systems in Silicon*, pp. 22–29, Oct. 1997.
- [23] J. Kim and M. Papaefthymiou, "Block-based multi-period refresh for energy efficient dynamic memory," in *14th Annual IEEE International ASIC/SOC Conference*, pp. 193–197, 2001.
- [24] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, "Raidr: Retention-aware intelligent dram refresh," in *39th Annual International Symposium on Computer Architecture (ISCA)*, pp. 1–12, Jun. 2012.
- [25] R. Venkatesan, S. Herr, and E. Rotenberg, "Retention-aware placement in dram (rapid): software methods for quasi-non-volatile dram," in *The Twelfth International Symposium on High-Performance Computer Architecture*, pp. 155–165, Feb. 2006.



- [26] M. Mutyam and V. Narayanan, "Working with process variation aware caches," in *Proceedings of the Conference on Design, automation and test in Europe*, pp. 1152–1157, 2007.
- [27] H. Kalter, C. Stapper, J. Barth, J.E., J. DiLorenzo, C. Drake, J. Fifield, J. Kelley, G.A., S. Lewis, W. van der Hoeven, and J. Yankosky, "A 50-ns 16-mb dram with a 10-ns data rate and on-chip ecc," in *IEEE Journal of Solid-State Circuits*, vol. 25, no. 5, pp. 1118–1128, Oct. 1990.
- [28] D. Weiss, J. Wu, and V. Chin, "The on-chip 3-mb subarray-based third-level cache on an itanium microprocessor," in *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1523–1529, Nov. 2002.
- [29] 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 *International Symposium on Low Power Electronics and Design*, pp. 207–212, 2001.
- [30] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9-v, 150-mhz, 10-mw, 4 mm<sup>2</sup>, 2-d discrete cosine transform core processor with variable threshold-voltage (vt) scheme," in *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1770–1779, Nov. 1996.
- [31] S. Thompson, I. Young, J. Greason, and M. Bohr, "Dual threshold voltages and substrate bias: Keys to high performance, low power, 0.1  $\mu\text{m}$  logic designs," in *Symposium on VLSI Technology*, pp. 69–70, Jun. 1997.
- [32] 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, "1.1v 1ghz communications router with on-chip body bias in 150nm cmos," in *IEEE International Solid-State Circuits Conference*, vol. 2, pp. 218–482, 2002.
- [33] Y. Oowaki, M. Noguchi, S. Takagi, D. Takashima, M. Ono, Y. Matsunaga, K. Sunouchi, H. Kawaguchiya, S. Matsuda, M. Kamoshida, T. Fuse, S. Watanabe, A. Toriumi, S. Manabe, and A. Hojo, "A sub-0.1  $\mu\text{m}$  circuit design with substrate-over-biasing [cmos logic]," in *IEEE International Solid-State Circuits Conference*, pp. 88–89, 420, Feb. 1998.
- [34] M. Miyazaki, G. Ono, T. Hattori, K. Shiozawa, K. Uchiyama, and K. Ishibashi, "A 1000-mips/w microprocessor using speed adaptive threshold-voltage cmos with forward bias," in *IEEE International Solid-State Circuits Conference*, pp. 420–421, 475, 2000.
- [35] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," in *IEEE International Solid-State Circuits Conference*, vol. 1, pp. 422–478 vol.1, 2002.

- [36] C. Lam, "Cell design considerations for phase change memory as a universal memory," in *International Symposium on VLSI Technology, Systems and Applications*, pp. 132–133, Apr. 2008.
- [37] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting phase change memory as a scalable dram alternative," in *Proceedings of the 36th Annual International Symposium on Computer Architecture*, pp. 2–13, 2009.
- [38] J. Bruce, S. W., and D. Wang, *Memory Systems Cache, DRAM, Disk*. San Fransisco: Morgan Kaufmann, 2008.
- [39] R. Ng, "Memory-fast computer memories," in *IEEE Spectrum*, vol. 29, no. 10, pp. 36–39, Oct. 1992.
- [40] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*. New York: Springer Science, 2005.
- [41] S. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Var-ius: A model of process variation and resulting timing errors for microarchitects," in *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 3–13, Feb. 2008.
- [42] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling within-die spatial correlation effects for process-design co-optimization," in *Sixth International Symposium on Quality of Electronic Design*, pp. 516– 521, Mar. 2005.
- [43] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 4, pp. 619–631, Apr. 2007.
- [44] Z. Zhang-Ming, W. Da-Jing, Y. Yin-Tang, and E. Yun-Fei, "A statistical rcl interconnect delay model taking account of process variation," in *Chinese Physics*, vol. 20, no. 1, Mar. 2011.
- [45] S.-C. Wong, G.-Y. Lee, and D.-J. Ma, "Modeling of interconnect capacitance, delay, and crosstalk in vlsi," in *IEEE Transactions on Semiconductor Manufacturing*, vol. 13, no. 1, pp. 108–111, Feb. 2000.
- [46] X. Qi, G. Wang, Z. Yu, R. Dutton, T. Young, and N. Chang, "On-chip inductance modeling and rlc extraction of vlsi interconnects for circuit simulation," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 487–490, 2000.
- [47] S. Kulkarni, D. Sylvester, and D. Blaauw, "A statistical framework for post-silicon tuning through body bias clustering," in *IEEE/ACM International Conference on Computer-Aided Design*, pp. 39–46, Nov. 2006.
- [48] B. Hargreaves, H. Hult, and S. Reda, "Within-die process variations: How accurately can they be statistically modeled?" in *Asia and South Pacific Design Automation Conference*, pp. 524–530, Mar. 2008.

- [49] A. Srivastava, S. Shah, K. Agarwal, D. Sylvester, D. Blaauw, and S. Director, “Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance,” in *Proceedings of 42nd Design Automation Conference*, pp. 535–540, Jun. 2005.