I, Romana Fernandes, hereby submit this original work as part of the requirements for the degree of:

Master of Science

in Electrical Engineering

It is entitled:

Estimation and Optimization of Leakage Power in the Presence of Process Variations

Student Signature: Romana Fernandes

This work and its defense approved by:

Committee Chair: Ranganadha Vemuri, PhD

Wen-Ben Jone, PhD

Carla Purdy, C, PhD

2/5/2010
Estimation and Optimization of Leakage Power in the Presence of Process Variations

A thesis submitted to the
Division of Research and Advanced Studies
of the University of Cincinnati
in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

in the Department of Electrical and Computer Engineering
of the College of Engineering

by

Romana Fernandes
B.E. Electronics and Communication Engineering,
PES Institute of Technology, Bangalore, India, 2004.

Thesis Advisor and Committee Chair: Dr. Ranga R. Vemuri

February 4, 2010
Abstract

With the increasing importance of run-time leakage power dissipation (around 55% of total power), it has become necessary to accurately estimate it not only as a function of input vectors, but also as a function of process parameters. In this work, the importance of considering the effects of process parameter variations for the accurate estimation of leakage power is emphasized and supported by experimental results.

Leakage power corresponding to the maximum vector presents itself as a higher bound for run-time leakage and is a measure of reliability. This problem is addressed and a heuristic is developed and implemented to accurately estimate the probabilistic distribution of the maximum run-time leakage power in the presence of variations in process parameters such as threshold voltage, critical dimensions and doping concentration. Both sub-threshold and gate leakage current have been considered. A heuristic approach is proposed to determine the vector that causes the maximum leakage power under the influence of random process variations. This vector is then used to estimate the lognormal distribution of the total leakage current of the circuit by summing up the lognormal leakage current distributions of the individual standard cells at their respective input levels. The proposed method has been effective in accurately estimating the leakage mean, standard deviation and probability density function of ISCAS-85 benchmark circuits.

Run-time leakage power is becoming a dominant component of the total power consumption of a CMOS circuit. A fast and accurate method for the estimation of average run-time leakage power using input signal probabilities is implemented. The proposed method considers signal correlations due to re-convergent fan-out nodes and process variations to improve the accuracy. The heuristic developed, estimates the average leakage power distribution by computing the mean and standard deviation. This heuristic was tested on ISCAS-85 benchmark circuits and was verified for accuracy.

An optimization technique to minimize the average run-time leakage power using a dual threshold voltage approach is implemented. The low leakage variability of high threshold devices helps reduce the high variability of nominal threshold devices and makes leakage reduction possible to a large extent. A significant improvement was seen for mean and standard deviation when tested on ISCAS-85 benchmark circuits.
Acknowledgment

It gives me immense pleasure to thank everyone who has helped me during the course of my Masters studies at the University of Cincinnati.

I am thankful to God for showering his blessings on me and for being my silent source of comfort. I want to take this opportunity to express my gratitude to my mom and dad for encouraging me and for always being there for me. I am grateful for their efforts and their unconditional love for me. My sisters Roshni and Nikita have always supported me in all my endeavors and I would like to thank them for being there for me.

I am grateful to Dr. Ranga Vemuri, my thesis advisor for his valuable guidance and encouragement. I learnt a lot under his guidance and he has been a great teacher to me. I am also thankful to all my professors at the University of Cincinnati for the knowledge I received here. In particular, I would like to extend my thanks to Dr. Wen-Ben Jone and Dr. Carla Purdy for taking time off their busy schedules to be on my defense committee.

I would like to take this opportunity to thank my dear friends at the University of Cincinnati, especially Thiaga, Aditi and her family for their friendship and making my stay here a memorable one. I would also like to remember my good old friends especially, Seena, Devi, Loretta, CV, Vedank, Arun and Vineeth for their friendship and moral support. Special thanks to my friends at DDEL, especially Shubhankar, Hao, Surya, Pritesh, Annie and Mike for their help when I needed the most. Thanks to my room-mates, Sasmita, Asha and Lily for supporting me and making my stay a pleasant one.

Finally, I would like to thank all my well-wishers, for wishing me the very best.
Contents

1 Introduction ................................................................. 1
  1.1 Motivation .............................................................. 1
  1.2 Background ............................................................ 3
    1.2.1 Design for Manufacturability .................................... 3
    1.2.2 Dependence of Leakage Power on Process Parameters ....... 5
    1.2.3 Leakage Power Classification ..................................... 8
    1.2.4 Previous Work in Statistical Leakage Analysis ............... 8
  1.3 Hspice and Monte Carlo Experiments ............................... 10
    1.3.1 Experiments to Demonstrate the Dependence of Leakage on Process Parameters ... 10
    1.3.2 Experiment to Show the Variability of Leakage Power with Process Variations ... 11
  1.4 Problem Definition .................................................. 12
  1.5 Thesis Outline ........................................................ 14

2 Estimation of Maximum Leakage Power Bound ....................... 15
  2.1 Methods to Determine the Maximum / Minimum Leakage Vector .... 15
  2.2 Motivation - Dependence of Leakage on Inputs in the Presence of Process Variations ... 16
  2.3 Maximum Leakage Vector in the Presence of Process Variations .... 17
    2.3.1 Definitions ...................................................... 18
    2.3.2 Heuristic to Determine the Maximum Leakage Vector .......... 23
  2.4 Estimation of Total Leakage Current Distribution ................ 23
    2.4.1 Sum of Log-normal Probability Distributions .................. 25
    2.4.2 Overall Approach ................................................. 26
2.5 Results ............................................................................................................................................... 26
   2.5.1 Leakage Power Distribution Corresponding to the Maximum Leakage Vector in the Presence of Variations ........................................................................................................................................... 27
   2.5.2 Pessimistic Approach .................................................................................................................. 31
   2.5.3 Maximum Leakage Vector in the Presence of Variations ............................................................. 32
   2.5.4 Complexity and Usage ............................................................................................................... 32
   2.5.5 Determination of Minimum Leakage Power Vector ..................................................................... 33

3 Average Runtime Leakage Power Estimation with Process Variations ............................................. 36
   3.1 Dependence of Average Leakage Power on Input Signal Probabilities ............................................ 36
   3.2 Average Leakage Estimation and Optimization Flow ........................................................................ 37
   3.3 Significance of Considering Run-time Signal Probabilities ............................................................... 37
   3.4 Computation of Node Signal Probabilities Considering Correlations ............................................ 40
      3.4.1 Zero Algorithm ......................................................................................................................... 40
      3.4.2 Weighted Averaging Algorithm ............................................................................................... 40
      3.4.3 Dynamic Weighted Averaging Algorithm (DWAA) ................................................................. 41
      3.4.4 Illustration of Dynamic Weighted Averaging Algorithm .......................................................... 44
   3.5 State Probability Based Average Leakage Power ............................................................................. 45
   3.6 Sum of Log-normal Distributions .................................................................................................. 46
   3.7 Overall Approach to Estimate the Average Leakage Power ............................................................ 47
      3.7.1 Illustration of Overall Approach .............................................................................................. 47
   3.8 Results ............................................................................................................................................ 48

4 Optimization of Average Runtime Leakage Power in the Presence of Process Variations ............... 51
   4.1 Optimization Techniques for Leakage Power Reduction ............................................................... 51
      4.1.1 Input Vector Control .................................................................................................................. 51
      4.1.2 Dual Threshold Voltage Technique ......................................................................................... 52
      4.1.3 Channel Length Biasing .......................................................................................................... 52
      4.1.4 Power Supply Gating .............................................................................................................. 52
      4.1.5 Body Biasing ........................................................................................................................... 52

vii
List of Figures

1.1 Catastrophic Defect - Short and Open [Source: [56]] ........................................... 4
1.2 Parametric Defect - Channel Length Variations ..................................................... 4
1.3 Systematic Defect - Dishing and Erosion [Source: [47]] ....................................... 4
1.4 Trends in Process Parameters [Source: [42]] ......................................................... 5
1.5 Timing and Leakage Power Variations [Source: [15]] ........................................... 7
1.6 OFF State Leakage [2] ............................................................................................. 7
1.7 ON State Leakage [2] .............................................................................................. 7
1.8 Runtime Leakage Power [Source: [37]] ................................................................. 9
1.9 Dependence on NMOS $L_{ch}$ ................................................................................ 11
1.10 Dependence on PMOS $L_{ch}$ ................................................................................ 11
1.11 Dependence on NMOS $V_{th}$ ................................................................................ 11
1.12 Dependence on PMOS $V_{th}$ ................................................................................ 11
1.13 Dependence on NMOS $N_{ch}$ .............................................................................. 12
1.14 Dependence on PMOS $N_{ch}$ .............................................................................. 12
1.15 Dependence on NMOS $W$ .................................................................................. 12
1.16 Dependence on PMOS $W$ .................................................................................. 12
1.17 Dependence on NMOS $T_{ox}$ .............................................................................. 13
1.18 Dependence on PMOS $T_{ox}$ .............................................................................. 13
1.19 Leakage Power Distribution - C1355 ................................................................. 14
2.1 Example Circuit .................................................................................................... 17
2.2 Illustrative Example ............................................................................................. 18
2.3 Comparison of Heuristic Mean with Verified Mean ............................................ 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4 Comparison of Heuristic Standard Deviation with Verified Standard Deviation</td>
<td>28</td>
</tr>
<tr>
<td>2.5 Comparison of Heuristic Mean with Monte Carlo Mean</td>
<td>28</td>
</tr>
<tr>
<td>2.6 Comparison of Heuristic Mean with Leakage Current Obtained Using [46]</td>
<td>29</td>
</tr>
<tr>
<td>2.7 Comparison of Heuristic Pessimistic Maximum with Verified Maximum</td>
<td>29</td>
</tr>
<tr>
<td>2.8 Comparison of Heuristic Pessimistic Maximum with the Leakage Obtained Using [46]</td>
<td>30</td>
</tr>
<tr>
<td>2.9 Comparison of Heuristic Mean with Leakage Obtained Using the Vector from [46]</td>
<td>30</td>
</tr>
<tr>
<td>3.1 Average Leakage Estimation and Optimization</td>
<td>38</td>
</tr>
<tr>
<td>3.2 State Probabilities of all Nodes in Benchmark Circuit (P_Is = 0.5) [Source: [36]]</td>
<td>39</td>
</tr>
<tr>
<td>3.3 State Probabilities of all Nodes in Benchmark Circuit (P_Is = 0.2 or 0.8) [Source: [36]]</td>
<td>39</td>
</tr>
<tr>
<td>3.4 Example Circuit</td>
<td>44</td>
</tr>
<tr>
<td>3.5 Comparison of Heuristic Mean with Verified Mean</td>
<td>50</td>
</tr>
<tr>
<td>3.6 Comparison of Heuristic Standard Deviation with Verified Standard Deviation</td>
<td>50</td>
</tr>
<tr>
<td>4.1 Optimized Leakage Mean Results</td>
<td>56</td>
</tr>
<tr>
<td>4.2 Optimized Leakage Standard Deviation Results</td>
<td>57</td>
</tr>
<tr>
<td>A.1 Monte Carlo Analysis in Hspice [Source:[4]]</td>
<td>69</td>
</tr>
</tbody>
</table>
List of Tables

1.1 Nominal Values of Process Parameters in 32nm Technology .......................... 6
1.2 Variability Data for 32nm technology ....................................................... 6

2.1 Notations ................................................................................................. 18
2.2 Leakage Mean and Standard Deviation of Standard Cells ......................... 19
2.3 Determination of CCO, CC1, CC0 list and CC1 list ................................. 20
2.4 Determination of PWIC ........................................................................... 21
2.5 Determination of PWLA .......................................................................... 21
2.6 Conflicting and Dominated Cells ............................................................. 22
2.7 Determination of Cell Cost ....................................................................... 22
2.8 Updated Values ......................................................................................... 25
2.9 Comparison of Mean and Standard Deviation Obtained Using the Heuristic (H) with Random Vector Testing (R) ................................................................. 31
2.10 Comparison of Heuristic Mean with Monte Carlo Mean ......................... 31
2.11 Comparison of Mean Obtained Using Heuristic Approach (H) with the Leakage Current Obtained Using [46] ................................................................. 32
2.12 Maximum Sampled Leakage of Standard Cells ....................................... 32
2.13 Comparison of Maximum Leakage Obtained Using the Pessimistic Approach with Random Vector Testing ................................................................. 33
2.14 Comparison of Leakage Obtained by the Pessimistic Approach (P) with Leakage Obtained Using [46] ................................................................. 34
2.15 Comparison of Leakage Means Using the Max Vector Obtained by Heuristic Approach (H) with the Max Vector Obtained by [46] ................................. 35
Chapter 1

Introduction

Technology scaling into the deep nanometer geometries has seen an increase in process variability due to lithographic inaccuracies and this problem magnifies with every generation. Though the advantages of scaling are evident and it is necessary to continue with this trend, it is also important to consider variability data in order to make accurate estimations of performance measures in order to improve the parametric yield of a design after fabrication. Thus, it has become highly important to accurately estimate performance measures such as timing, leakage power and noise in the design phase itself.

Among the three important performance measures - timing, power and noise, power estimation and in particular leakage power estimation is gaining tremendous importance with scaling. With the increase of leakage power contribution towards the total power of a circuit, it has become highly important to estimate the leakage power in the presence of process variations. The traditional methods of leakage power estimation, such as nominal analysis, underestimate leakage power by a large margin whereas corner analysis, on the other hand, overestimates leakage power dissipation. Statistical leakage power estimation on the other hand gives an accurate estimate while considering the variations in process parameters. Therefore, statistical leakage power estimation and optimization is gaining tremendous importance in the current technologies and will continue to do so with further scaling.

1.1 Motivation

Variations in process parameters such as channel length ($L_{ch}$), threshold voltage ($V_{th}$), channel doping concentration ($N_{ch}$), gate oxide thickness ($T_{ox}$) and gate width ($W_{gate}$) are the major contributors to variability
in timing and leakage power. It has been shown that this dependence on process parameters is linear in the case of timing [7] and exponential in the case of leakage power [45]. It has also been shown that a 30% variation in circuit frequency induced by parameter variations can result in nearly 20X variation in leakage power [15]. Due to this high variability in leakage power, it becomes extremely important to consider the effects of variability in the estimation techniques for sufficient accuracy. Traditional estimation techniques like estimation at the nominal values of the process parameters tend to under-estimate leakage where as corner analysis tends to overestimate leakage power. As a result of under-estimation, there may be a failure in meeting the power yield and this may also cause reliability issues in chips that do meet the power yield during their life span. Over-estimation on the other hand may result in over-design in the form of designing unnecessary guard bands / rings and failure to meet timing specifications [12]. Therefore, it is extremely important to accurately estimate leakage power as a function of process variations.

A chip can settle in an idle state for a considerable amount of time or can enter stand-by mode several times during its life-span. It enters an idle state or stand-by mode with a different set of input vectors each time. In this scenario, it is not assured that the leakage power specifications are met each time. Excessive leakage power dissipation for a long period of time results in a drastic increase in the thermal profile of most chips during their operation and makes them susceptible to failure. Maximum leakage power estimation provides an upper bound and guarantees that the design constraints are met irrespective of the circuit input state [14]. Maximum leakage can also be used to find hot-spots in a physical design. It can also be used to estimate the worst case battery life of a portable device [14].

Traditionally, leakage power was considered important only in the stand-by mode whereas dynamic power was considered important in the active mode of operation. But, due to the shrinking physical dimensions, the contribution of dynamic power to the total power has reduced with the growth of leakage power [49]. With its increasing importance, leakage contribution during run-time has become a deciding factor to determine the maximum power bound for reliability of a chip and as a measure of its ever increasing thermal profile. This leakage power is called run-time leakage power dissipation [37]. Runtime leakage is vector dependent and changes each time the input vector changes.

It has been well established that the maximum leakage of a circuit can be greater by a few orders of magnitude than the minimum leakage and is dependent on the input vectors associated with them [26]. But, this dependence of maximum leakage power on input vectors alone is not correct anymore. In the presence
of variability, the input vectors that cause the maximum or the minimum leakage current changes. Hence, maximum leakage now depends not only on the input vector but also on process parameter variations.

Leakage current comprises of several components [48] of which sub-threshold and gate leakage are the prominent ones [39]. With leakage power contributing more than 55% of the total power in present day technologies (32nm and beyond) and this trend being predicted to only increase in future technologies [5],[16], accurate estimation methods considering all the important types of leakage has become a necessity.

Due to this increased contribution of leakage power to total power, it is important to estimate the average leakage power dissipated during the run-time / active mode of a circuit.

This research work presents techniques to estimate the maximum leakage power bound and the average run-time leakage power distribution as a function of both process variations and input vectors. An optimization technique is also presented to reduce the estimated statistical average leakage power.

The next few sections provide some insight into the background for this work. It also formally defines the problem and gives an outline of this research work.

1.2 Background

1.2.1 Design for Manufacturability

Design for manufacturability is defined as design procedures that aim at maximizing the parametric yield of VLSI chips. Imperfections / variations are caused in device dimensions and doping concentration because of immature manufacturing processes that have not kept up with the rapid leaps in technology migration.

The inaccuracies caused during fabrication give rise to defects. Defects can be of three types -

a. Catastrophic defects
b. Parametric defects
c. Systematic defects

While catastrophic defects reduce the functional yield of a design (the number of manufactured chips that function correctly), the parametric defects reduce the parametric yield of a design (the number of manufactured chips that meet performance, power and noise constraints). Examples of catastrophic defects include opens and shorts caused due to metal migration. Figure 1.1 shows examples of catastrophic defects on metal
layers. Examples of parametric defects include variability in device dimensions and non-uniform doping concentration. Figure 1.2 shows an example of parametric defect - variations in channel length.

Systematic defects are layout / physical design dependent and affect devices that are close to each other. Systematic defects include dishing and erosion which are caused during Chemical Mechanical Planarization as shown in Figure 1.3.

In present day technologies, a majority of chips do not meet the parametric yield of a design after fabrication. They fail to meet the timing or leakage constraints imposed on them and hence do not make it to the market. The most important cause for the reduction in parametric yield is process variations. These variations are caused due to lithographic inaccuracies which result in deviations from the nominal values of important process parameters such as channel length ($L_{ch}$), threshold voltage ($V_{th}$), gate oxide thickness ($T_{ox}$) and
gate width ($W_{gate}$). Non-uniformity in channel doping concentration ($N_{ch}$) during the doping process is also a cause of parametric defect. The trend of variability data is increasing with technology scaling as shown in Figure 1.4 [42]. The nominal values for 32nm technology parameters are given by Table 1.1. The ITRS variability data for a present day technology (32nm) is given by Table 1.2. It has been shown that such high variability in process parameters causes high variation in performance and power. A 30% variation in circuit frequency induced by parameter variations can result in nearly 20X variation in leakage power [15] as shown in Figure 1.5.

### 1.2.2 Dependence of Leakage Power on Process Parameters

Leakage power is synonymous with leakage current as leakage power is the product of VDD and leakage current. Therefore, the two terms are used interchangeably in this work. Leakage current is composed of several components. The important components of leakage power are -

a. Sub-threshold leakage

b. Gate leakage
### Table 1.1: Nominal Values of Process Parameters in 32nm Technology

<table>
<thead>
<tr>
<th>Process Parameters</th>
<th>Nominal Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMOS $L_{ch}$</td>
<td>32nm (2$\lambda$)</td>
</tr>
<tr>
<td>PMOS $L_{ch}$</td>
<td>32nm (2$\lambda$)</td>
</tr>
<tr>
<td>NMOS $W_{gate}$</td>
<td>112nm (7$\lambda$)</td>
</tr>
<tr>
<td>PMOS $W_{gate}$</td>
<td>224nm (14$\lambda$)</td>
</tr>
<tr>
<td>NMOS $T_{oxe}$</td>
<td>1.65nm</td>
</tr>
<tr>
<td>PMOS $T_{oxe}$</td>
<td>1.75nm</td>
</tr>
<tr>
<td>NMOS $V_{th}$</td>
<td>0.5088V</td>
</tr>
<tr>
<td>PMOS $V_{th}$</td>
<td>0.4500V</td>
</tr>
<tr>
<td>NMOS $N_{ch}$</td>
<td>4.12e+18</td>
</tr>
<tr>
<td>PMOS $N_{ch}$</td>
<td>3.07e+18</td>
</tr>
</tbody>
</table>

### Table 1.2: Variability Data for 32nm technology

<table>
<thead>
<tr>
<th>Process Parameters</th>
<th>Variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical Dimensions (Lch, Wgate, Tox)</td>
<td>12%</td>
</tr>
<tr>
<td>Threshold Voltage (Vth) (Including doping variability)</td>
<td>58%</td>
</tr>
</tbody>
</table>

- **c. Band to band tunneling leakage**
- **d. Gate induced drain leakage**

**a. Sub-threshold leakage** - The off-state leakage current of a MOSFET that flows from source to drain is defined as the sub-threshold leakage current.

**b. Gate leakage** - The on-state leakage component of a MOSFET that flows from the channel to the gate due to thinner and thinner gate oxides is defined as gate leakage.

**c. Band to band tunneling leakage (BTBT)** - The leakage component caused by electron tunneling across the reverse biased p-n junction between drain/source and substrate of a MOSFET is defined as BTBT leakage.

**d. Gate induced drain leakage (GIDL)** - The drain to substrate leakage in the OFF state of a MOSFET caused by high electric field is defined as GIDL.

Among the various leakage components, two of them dominate leakage power - sub-threshold ($I_{sub}$) and gate leakage ($I_{gate}$) currents. Sub-threshold leakage is the most dominant leakage mechanism occurring...
in the OFF state of a transistor and gate leakage is the most dominant leakage mechanism in its ON state [2]. The OFF and ON state leakage components are shown in Figure 1.6 and Figure 1.7 respectively. Considering both these types of leakage and the interactions between them, the total leakage of a circuit can be approximated using Equation 1.1.

\[ I_{\text{total}} = I_{\text{sub}} + I_{\text{gate}} \] (1.1)

\( I_{\text{sub}} \) and \( I_{\text{gate}} \) vary with variation in process parameters such as channel length \( (L_{\text{ch}}) \), threshold voltage \( (V_{\text{th}}) \), channel doping concentration \( (N_{\text{ch}}) \), gate oxide thickness \( (T_{\text{ox}}) \) and gate width \( (W_{\text{gate}}) \). The BSIM equation for sub-threshold leakage current is given by Equation 1.2 [2]. The threshold voltage is given by Equation 1.3 and the dependence of \( V_{\text{th}} \) on \( L_{\text{ch}} \) is given by Equation 1.4 [11].

\[ I_{\text{sub}} = I_0 \exp \left( \frac{V_{gs} - V_{\text{th}}}{\eta V_t} \right) \left[ 1 - \exp \left( \frac{-V_{ds}}{V_t} \right) \right] \] (1.2)

where, \( I_{\text{sub}} \) is the sub-threshold leakage current, \( I_0 \) is the saturation current, \( V_{gs} \) is the gate to source voltage,
$V_{ds}$ is the drain to source voltage, $V_{th}$ is the threshold voltage, $V_t = kT/q$ is the thermal equivalent voltage, $k$ is the Boltzmann constant, $T$ is the temperature and $q$ is the charge on an electron.

$$V_{th} = V_{fb} + |2\phi_p| + \frac{\lambda_b}{C_{ox}} \sqrt{2qN_{ch}\varepsilon_s(|2\phi_p| + V_{sb})} - \lambda_d V_{ds}$$

(1.3)

$$V_{th,i} = V_{th0} + 0.05 - V_{dd} e^{-\delta L_{e,i}}$$

(1.4)

where, $V_{fb}$ is the flat band voltage, $\phi_p$ is the surface potential, $\lambda_b$ is the body-effect factor, $\lambda_d$ is the DIBL co-efficient, $V_{sb}$ is source to body voltage, $N_{ch}$ is the channel doping concentration.

The equation for gate leakage current is given by Equation 1.5 [16].

$$I_{gate} = W A_g \left( \frac{V_{dd}}{T_{ox}} \right)^2 \left[ \exp \left( -B_b \frac{T_{ox}}{V_{dd}} \right) \right]$$

(1.5)

where, $A_g$ and $B_b$ are physical constants.

### 1.2.3 Leakage Power Classification

Leakage power can be classified into two types -

a. Static (standby) leakage power: The leakage power that is dissipated when the circuit is in stand-by mode. The input vectors are kept steady in this mode.

b. Active or Runtime leakage: The leakage power that is dissipated during the normal operation of the circuit. This has become a major concern in recent technologies.

Most of the work in leakage power analysis and minimization is being done to tackle stand-by power. With technology scaling, active leakage has become significant and needs to be addressed as well. Figure 1.8 shows the active leakage component during the normal operation of a circuit.

### 1.2.4 Previous Work in Statistical Leakage Analysis

Statistical estimation and optimization of leakage power has gained a lot of importance in the recent years. Some of the important work done in this area are briefly discussed in this section.
In [45], a method to estimate the leakage current variation is proposed by deriving an analytical expression for the probability density function for the same. Using this, the mean and variance for the entire circuit is found. The process parameters of interest in this research is the channel length variation. Both inter and intra die variations have been considered in this paper.

The work in [17] proposes a method to analyze the leakage power taking spatial correlations into consideration. An empirical model is derived for both sub-threshold and gate leakage currents taking state dependencies into account. The average leakage currents are summed up to derive the full-chip leakage distribution. Variations in channel length and thickness of gate oxide were considered in this work.

The effects of process ($L_{ch}$, $T_{ox}$), voltage (VDD) and temperature (T) have been considered in [58] for the full-chip estimation of subthreshold leakage current. It has further been shown in this work that accurate estimation is possible when temperature variations are considered due to the increase in leakage power due to electrothermal couplings between power and temperature.

Spatial correlations and their effects have been considered in [9] by using a notation of ‘cluster model’ for the computation of correlations. This method considers the effect of correlated channel length variation using layout information. The time complexity of the basic method is exponential but has been modified in this work to a quadratic time complexity algorithm for approximating leakage power variation.

In [20], an analytical modeling methodology has been proposed to estimate the leakage power of transistors and further been extended to estimate the leakage power of logic gates. Sub-threshold, gate and BTBT leakage power variations were considered in this method. Channel length variations have been considered
to develop the analytical models in this work.

The work in [10] represents leakage and propagation delay of a gate as quadratic polynomials to incorporate intra-die variations. The correlation factor is computed using Karhunen-Loeve Expansion. This has been extended to compute the total chip leakage power.

Statistical optimization of leakage power has been addressed in some of the recent work in [53], [13], [11] and [38]. In [53], statistical timing aware leakage optimization is done by using dual threshold devices and a gate sizing algorithm. Gate sizing is used in [13] to minimize leakage power statistically. The work in [11], reduces a function of mean and variance of leakage using a joint selection of gate sizes, gate length and threshold voltage using delay as a constraint.

1.3 Hspice and Monte Carlo Experiments

Two kinds of experimental results are shown in this section. The first set of experiments are hspice simulations that show the dependence of leakage on important process parameters. The second set of experiments are Monte Carlo simulations that show the variability in leakage because of the variability in process parameters.

1.3.1 Experiments to Demonstrate the Dependence of Leakage on Process Parameters

The spice experiments were performed on 32nm technology using predictive models [1]. The nominal values for the process parameters that were varied are given by Table 1.1. Typical variation data is shown in Table 1.2 [5]. Such significant variations in process parameters will induce large variations in performance and power.

Process parameters like NMOS and PMOS channel length, threshold voltage, channel doping concentration, gate width and thickness of gate oxide were individually swept in hspice and simulations were performed. It was observed that the leakage current of ISCAS-85 benchmark circuit, C1355 with 546 gates was dependent on the process parameters given by Equations 1.6 and 1.7. This agrees with the analytical expressions and previous work in [45],[52],[17],[11]. The results of the hspice simulations done to show the relationship between leakage power and process parameters are given by Figures 1.9 - 1.18.

\[ I_{sub} \propto \frac{1}{e^{L_{ch}}}, \quad I_{sub} \propto \frac{1}{e^{V_{th}}}, \quad I_{sub} \propto \frac{1}{N_{ch}}, \quad I_{sub} \propto W \] (1.6)
1.3.2 Experiment to Show the Variability of Leakage Power with Process Variations

500 Monte Carlo simulations were performed on an example ISCAS-85 benchmark circuit, C1355 with 546 gates by varying all the parameters of interest ($L_{ch}$, $V_{th}$, $N_{ch}$, $W_{gate}$ and $T_{ox}$) in a Gaussian manner. The variability data from table 1.2 [5] was used in the experiments. The result of this experiment is shown in Figure 1.19. It was observed in this experiment that the mean of the distribution is almost four times the nominal value for leakage current in the case of this benchmark circuit and the maximum bound is almost 8 times the nominal value.
Hence, from these experiments, it can be concluded that due to the dependence of leakage current represented by Equations 1.6 and 1.7, leakage current can no longer be represented by a single nominal or a corner case value. It is in fact a log-normal probabilistic distribution with mean $\mu$ which is significantly greater than the nominal value and has standard deviation $D$ [45].

### 1.4 Problem Definition

It was shown in the earlier sections that leakage power is becoming dominant in the deep sub-micron processes and the effects of process variability can no longer be ignored. In the present day technologies, leakage is no longer important only in the stand-by mode, but is also important during run-time and hence is vector dependent. Considering all the afore mentioned reasons, it can be concluded that leakage power
is a function of both input vectors and process variations apart from being dependent on the technology and circuit functionality. For this reason, it is attempted in this work to accurately estimate and optimize leakage power considering all the factors that leakage is dependent on. The formal definition of the problem is divided into three sub-problems -

a. **Accurate Estimation of Maximum Vector Dependent Leakage Power in the Presence of Process Variations**: The goal of the first technique is to estimate the maximum leakage power bound of a circuit in the presence of process variations. This method can also be extended to determine the minimum leakage vector and the leakage associated with it. The minimum vector thus obtained, can be used as a sleep vector in standby mode.

b. **Estimation of Average Run-time Leakage Power Considering Correlated Signal Probabilities in the Presence of Process Variations**: The goal of the second approach is to estimate the average run-time leakage power of a circuit, given the signal probabilities of the primary inputs. This approach considers correlations between signals due to re-convergent fan-out nodes and variations in process parameters for increasing the accuracy of estimation.

c. **Optimization of Average Run-time Leakage Power**: Finally, an optimization technique which aims at reducing the average run-time leakage power of a circuit with timing as a constraint is proposed.
1.5 Thesis Outline

Chapter 2 presents the approach developed to determine the maximum leakage power bound for a circuit. It presents the motivation and experiments for this problem. It explains the heuristic developed to determine the maximum leakage vector in the presence of process variations. This vector is used to determine the maximum leakage power bound and the corresponding probabilistic distribution function of leakage power. Finally, the results of the implemented technique are also presented.

Chapter 3 presents the approach developed to estimate the average run-time leakage power of a circuit. The method used to determine the correlated signal probabilities is discussed first. This is followed by the heuristic to determine the average run-time leakage power in the presence of process variations. The results obtained using this heuristic are also presented in this chapter.

Chapter 4 presents an optimization technique used to reduce the average run-time leakage power of a circuit. The optimization technique which uses timing constraints in order to achieve leakage minimization is explained. Finally, the results of leakage reduction for a pre-determined timing penalty are also presented.

Chapter 5 discusses the concluding remarks and future work.
Chapter 2

Estimation of Maximum Leakage Power Bound

2.1 Methods to Determine the Maximum / Minimum Leakage Vector

The computation of the minimum leakage vector / maximum leakage vector has been dealt with in the past. A brief summary of these methods are presented in this section. However, the techniques described in this section, do not take process variations into consideration. The different techniques that have been developed to address this problem can be summarized as follows.

a. Random Vectors Technique: A near exhaustive simulation involving thousands of vectors is performed to determine the vector that causes the minimum or maximum leakage power. The vectors are chosen in a pseudo random manner. In this technique, it is attempted to cover most of the input sample space. For a circuit with a large number of primary inputs (‘N’), the total number of input vectors that need to be simulated grow exponentially and is equal to $2^N$. Such a large number of primary inputs in cases where ‘N’ is sufficiently large is prohibitive in terms of simulation time. This approach has been proposed in [30],[18]. The work in [18] uses a genetic algorithm technique to select the random vectors.

b. Integer Linear programming: In integer linear programming (ILP), a linear objective function is formulated with a set of constraints [41]. The objective function is chosen to be the sum of leakage power of all
the gates in a circuit. The decision variables in ILP are integers and the constraints grow linearly with circuit size. Work proposed in [24], [41] propose variations of ILP to determine the minimum stand-by leakage power of a CMOS circuit.

c. **Branch and Bound Technique**: In [14], a branch and bound technique is used to determine the maximum leakage vector. In this method, a constraint graph is formed based on the circuit structure and functionality of the components. Clique constraints, logic constraints and stem constraints are developed and used to create a constraint graph. An optimization performed on this constraint graph results in the maximum leakage vector. Another variation of branch and bound technique has been developed in [32].

d. **SAT Solver**: The work proposed in [8] uses a pseudo boolean solver (PBS) to solve the problem of minimum / maximum leakage vector determination. The circuits are represented in Conjunctive Normal Form (CNF) and an objective constraint indicates the amount of targeted leakage. Using these constraints, the solution space is incrementally explored and the objective is achieved.

e. **BDD Technique**: Binary decision diagrams and its variants are used in [27] and [19] in order to determine the minimum / maximum leakage vector. The approach in [19] is based on “Implicit Enumeration of integer-valued decision diagrams”.

2.2 **Motivation - Dependence of Leakage on Inputs in the Presence of Process Variations**

Consider a CMOS inverter in 32nm technology, with nominal values of process parameters given by Table 1.1 [1]. This inverter has maximum leakage current when input state is ‘0’. However, when process parameters are changed to PMOS $L_{ch} = 1.837\lambda$ and NMOS $L_{ch} = 1.985\lambda$ with other parameters at their nominal values, the maximum leakage inducing input is ‘1’. In fact, when 500 Monte Carlo simulations vary the process parameters around their nominal values with variability given by Table 1.2 [5], it is observed that the mean leakage when the input vector is ‘1’ is far greater than the mean leakage when the input is ‘0’. In
fact, the mean leakage for input ‘1’ is almost double when compared to the mean leakage for input ‘0’. The reason for the inverter to change its maximum leakage state is because the \( I_{\text{sub}} \) of PMOS in its OFF state (input ‘1’) exceeds that of NMOS in its OFF state (input ‘0’) when the channel length of PMOS is less than the NMOS channel length. This is because of the increased sensitivity of PMOS sub-threshold current to channel length variations as compared to NMOS [58].

This example can be extended to a simple combinational circuit as shown in Figure 2.1. In the absence of process variations, the maximum leakage state for this circuit is ‘01’ whereas in the presence of variations, Monte Carlo simulations show that the input state for maximum leakage changes to ‘11’. This change in input vector changes the total leakage from \( \text{INV (0) + INV (1) + AND (101)} \) to \( 2*\text{INV (1) + AND (001)} \). Monte Carlo experiments have shown that, for larger cells, the maximum leakage vector differs from the one calculated using the conventional method in cases where the effects of process variations dominate the bias voltage effects or where there is increased sensitivity to a process parameter.

### 2.3 Maximum Leakage Vector in the Presence of Process Variations

Exact determination of the input vector that induces maximum (or minimum) leakage requires exhaustive simulations with all input vectors. Several approaches have been proposed to estimate this vector with reduced complexity focusing on sleep vectors for leakage minimization in stand-by mode [32], [6], [14]. Rao et al. [46] proposed an approach where the minimum (or maximum) leakage vector is determined by taking cell functionalities into account but the effects of process variations were not considered. This technique cannot be used in the presence of process variations, as demonstrated in Section 2.2. In this work, a heuristic is proposed to accurately estimate the maximum leakage vector considering both cell
functionalities as well as process variations [23]. The vector thus obtained is then used to determine the maximum sum leakage distribution of a circuit. This method can also be modified to find the minimum leakage vector and hence the minimum sum leakage distribution. Below, some of the definitions from Rao et al. provided in [46] are adapted to accommodate process variations. The definitions are also supported by an illustrative example given by Figure 2.2. Only the most dominant leakage state is considered for the purposes of illustration as opposed to the implemented algorithm which considers the top two maximum leakage states. This was done to keep the illustration simple. The notations used in the illustration are given by Table 2.1.

### 2.3.1 Definitions

It is assumed that the circuit under consideration can be decomposed into standard cells. A graph is constructed with cells as the nodes and nets as the edges. The leakage value associated with each state of a standard cell is calculated as a weighted sum of the mean leakage current and the standard deviation of that particular state as given by Equation 2.1. The mean and standard deviation are determined by performing 500 Monte Carlo simulations by varying the process parameters as shown in Tables 1.1 and 1.2 for each input combination for each cell. The mean and standard deviation of leakage for an inverter and a two input
Table 2.2: Leakage Mean and Standard Deviation of Standard Cells

<table>
<thead>
<tr>
<th>Cell</th>
<th>Nominal Value ((10^{-7}))</th>
<th>Mean ((10^{-7}))</th>
<th>Standard Deviation ((10^{-7}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV (Input '0')</td>
<td>1.49</td>
<td>2.25</td>
<td>5.26</td>
</tr>
<tr>
<td>INV (Input '1')</td>
<td>0.82</td>
<td>4.62</td>
<td>12.46</td>
</tr>
<tr>
<td>OR2 (Input '00')</td>
<td>2.29</td>
<td>7.79</td>
<td>14.41</td>
</tr>
<tr>
<td>OR2 (Input '01')</td>
<td>1.53</td>
<td>5.00</td>
<td>7.65</td>
</tr>
<tr>
<td>OR2 (Input '10')</td>
<td>0.92</td>
<td>2.52</td>
<td>4.54</td>
</tr>
<tr>
<td>OR2 (Input '11')</td>
<td>0.73</td>
<td>2.03</td>
<td>4.36</td>
</tr>
</tbody>
</table>

OR gate are given by Table 2.2. It was seen in Figure 1.19 that the probability density function of a circuit has an increased average leakage value given by its mean and a spread represented by the standard deviation. When \(\lambda\) is chosen to be equal to 1, the heuristic chooses cells with greater mean values but completely ignores the standard deviation and hence the spread of the leakage profile. When \(\lambda\) is chosen to be equal to zero, the mean leakage value is completely ignored and the heuristic chooses cells with greater spread. This could result in choosing cells with smaller means. Hence, \(\lambda\) is chosen to be 0.5 in the experiments to give equal importance to both leakage mean and standard deviation.

\[
P_i = \lambda \mu_i + (1 - \lambda) D_i
\]  

(2.1)

where, \(P_i\) is the probabilistic cell leakage in input state \(i\),
\(\mu_i\) is the mean leakage of the cell in input state \(i\),
\(D_i\) is the standard deviation of leakage in input state \(i\),
\(\lambda\), the weighting factor is a real number and \(\lambda \in [0, 1]\)

**Node Controllability:** The controllability of a node in a circuit is defined as the minimum number of inputs that have to be assigned to specific values in order to force the node output to a specific state. Every node is assigned two values, CC0 (controllability to force the cell output to 0) and CC1 (controllability to force the cell output to 1).

**Controllability List:** The constraints imposed on the primary input vector in order to force a node output to a specific state is defined as the controllability list. The two constraint lists associated with every node
Table 2.3: Determination of CC0, CC1, CC0 list and CC1 list

<table>
<thead>
<tr>
<th>Net</th>
<th>CC0 list - PI1 PI2 PI3 (CC0)</th>
<th>CC1 list - PI1 PI2 PI3 (CC1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>1XX (1)</td>
<td>0XX (1)</td>
</tr>
<tr>
<td>N2</td>
<td>X1X (1)</td>
<td>X0X (1)</td>
</tr>
<tr>
<td>N3</td>
<td>1XX (1)</td>
<td>0CX (C)</td>
</tr>
<tr>
<td>N4</td>
<td>XX0 (1)</td>
<td>XX1 (1)</td>
</tr>
</tbody>
</table>

output is CC0 list and CC1 list.

For example, in Figure 2.2, CC0 list of N3 would require either N1, N2 or PI2 to be ‘0’. A net which has the least fanout is chosen. In case of equal fanout nodes, a net is picked randomly and set to ‘0’. In this example N1=0 is chosen which requires PI=1 and PI2=X and PI3=X on the primary input lines. Table 2.3 gives the CC0 and CC1 lists for all internal nets in the example circuit.

**Probabilistic Worst Input Condition (PWIC):** The worst input condition for a cell represents the minimum number of primary inputs and their specific values that force the cell to its highest leakage in the presence of process variations.

It was observed in the experiments that every gate has two dominant worst leakage states. The remaining states dissipate far less leakage compared to these two dominant states. It was also observed that when the top two worst leakage states were considered the accuracy of the algorithm was improved. Hence, two Worst Input Conditions, PWIC1 and PWIC2 are defined, where PWIC1 yields the worst value for leakage and PWIC2 yields the second worst value.

For example, in the case of cell OR2, PWIC1=00 and PWIC2=01. To force the inputs of OR2 to PWIC1, N3 and N4 must be forced to ‘0’. This translates to CC0 of N3 & CC0 of N4 which is equal to 1XX & XX0 = 1X0. Table 2.4 gives the PWIC1 constraints for all the cells in the example circuit.

**Probabilistic Worst Leakage Advantage (PWLA):** If the PWIC of a cell cannot be satisfied, the cell may settle into one of its low leakage states. The increase in leakage when a cell is forced to its PWIC can be quantified by a metric called Probabilistic Worst Leakage Advantage. PWLA is given by the difference in the leakage of the worst leakage state and the average of the low leakage states and can be represented using Equation 2.2. Since two dominant PWICs are chosen, PWLAs are defined for each of them. Table 2.5 shows...
Table 2.4: Determination of PWIC

<table>
<thead>
<tr>
<th>Cell</th>
<th>PWIC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV-1</td>
<td>1XX</td>
</tr>
<tr>
<td>INV-2</td>
<td>X1X</td>
</tr>
<tr>
<td>AND3</td>
<td>0CX</td>
</tr>
<tr>
<td>BUF</td>
<td>XX1</td>
</tr>
<tr>
<td>OR2</td>
<td>1X0</td>
</tr>
</tbody>
</table>

Table 2.5: Determination of PWLA

<table>
<thead>
<tr>
<th>Cell</th>
<th>PWLA1 ($10^{-6}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV-1</td>
<td>0.957</td>
</tr>
<tr>
<td>INV-2</td>
<td>0.957</td>
</tr>
<tr>
<td>AND3</td>
<td>1.363</td>
</tr>
<tr>
<td>BUF</td>
<td>1.028</td>
</tr>
<tr>
<td>OR2</td>
<td>1.549</td>
</tr>
</tbody>
</table>

the PWLAs for the standard cells in the illustration.

\[
PWLA = PWIC - AVG(P_{LLS}) \tag{2.2}
\]

where, \(PWIC\) is the probabilistic cell leakage in PWIC,

\(P_{LLS}\) is the probabilistic cell leakage of low leakage states.

For example, in the case of cell OR2,

\[
PWLA_1 = P_{00} - 0.5(P_{11} + P_{10})
\]

\[
PWLA_2 = P_{01} - 0.5(P_{11} + P_{10})
\]

where, \(P_{ij}\) is the probabilistic leakage when the cell is in state \(ij\).

**Conflicting and Dominated Cells:** When the PWIC of a cell is satisfied, it will result in certain nodes in the circuit being forced to particular states because of the way the gates function. Cells \(C_i\) and \(C_j\) are said to be conflicting cells if they have opposing requirements for primary input \(P_m\). Cell \(C_i\) is said to dominate cell \(C_k\) if the input requirements for \(C_k\) to be in its worst input state is a subset of that required for cell \(C_i\) to be in its worst input state. Table 2.6 shows the conflicting and dominated cells for the illustrative example.

For example, OR2 and BUF in the illustration are conflicting cells because BUF requires PI3=1 to force it...
Table 2.6: Conflicting and Dominated Cells

<table>
<thead>
<tr>
<th>Cell</th>
<th>Conflicting Cells</th>
<th>Dominated Cells</th>
<th>Infeasible?</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV-1</td>
<td>AND3</td>
<td>-</td>
<td>NO</td>
</tr>
<tr>
<td>INV-2</td>
<td>-</td>
<td>-</td>
<td>NO</td>
</tr>
<tr>
<td>AND3</td>
<td>INV-1, OR2</td>
<td>-</td>
<td>YES</td>
</tr>
<tr>
<td>BUF</td>
<td>OR2</td>
<td>-</td>
<td>NO</td>
</tr>
<tr>
<td>OR2</td>
<td>AND3, BUF</td>
<td>INV-1</td>
<td>NO</td>
</tr>
</tbody>
</table>

Table 2.7: Determination of Cell Cost

<table>
<thead>
<tr>
<th>Cell</th>
<th>Cost ($\mu$A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV-1</td>
<td>-0.406</td>
</tr>
<tr>
<td>INV-2</td>
<td>0.957</td>
</tr>
<tr>
<td>AND3</td>
<td>LARGE NEG</td>
</tr>
<tr>
<td>BUF</td>
<td>-0.520</td>
</tr>
<tr>
<td>OR2</td>
<td>0.114</td>
</tr>
</tbody>
</table>

to its PWIC while OR2 requires PI3=0 to force it to its PWIC. On the other hand OR2 dominates INV-1 because satisfying PWIC1 of OR2 (1X0) would also mean satisfying the PWIC1 of INV-1 (1XX).

**Cost Function:** When the PWIC of a cell is satisfied, the PWICs of its conflicting cells are violated, while the PWICs of its dominated cells are satisfied. Therefore, the cost of satisfying the PWIC of a cell $C_i$ is calculated as given by $Cost(C_i)$. As a special case, costs of cells which have infeasible requirements for a primary input are assigned a large negative value, as they can never be forced to their PWIC. The costs of the standard cells in the illustration is given by Table 2.7.

$$Cost(C_i) = PWLA(C_i) + \sum(PWLA(DominatedCells(C_i))) - \sum(PWLA(ConflictingCells(C_i)))$$

For example, Cost(BUF) = PWLA(BUF)-PWLA(OR2). The costs associated with both the PWICs are calculated in this method.
2.3.2 Heuristic to Determine the Maximum Leakage Vector

This section gives the outline of HeuristicMax, the heuristic proposed to determine the maximum leakage vector in the presence of process variations. The heuristic is given by Algorithm 2.1. The input to the heuristic is the circuit which needs to be analyzed. The circuit is represented using a graph data structure with the standard cells represented by nodes and nets represented by edges in the graph. The first step in the heuristic is to sort the nets in increasing order from the primary inputs in order to reduce the overall complexity of the heuristic. The next step is to determine the controllability at 0 (CC0) and controllability at 1 (CC1) lists. These controllability lists are then used to compute the probabilistic worst input conditions (PWICs) and the probabilistic worst leakage advantage (PWLAs). The next step in the algorithm is to generate the conflicting and dominated cell lists for each cell in the circuit. Using the PWLAs, the cost function for each cell is computed. Cells with infeasible inputs are assigned a large negative value to indicate that their PWIC can never be satisfied. Iteratively, the PWIC of the cell with the maximum cost function is satisfied and the conflicting and dominated cell lists for all cells are updated appropriately until all the primary inputs are defined or when the cell list is empty. Finally, if some primary inputs remain undefined, such inputs are assigned either ‘0’ or ‘1’ depending on which input value puts the circuit into the maximum leakage state.

The complexity of determining the controllability and controllability lists in HeuristicMax was reduced by sorting the nets in the increasing order of its depth. A cell in the higher level was processed only after processing all the cells in the lower levels. By sorting, the need to back traverse all the way to the primary inputs was eliminated and hence the complexity was reduced.

In the illustrative example, INV-2 is selected and its input constraint - X1X is satisfied in the first iteration. The conflicting and dominated cell lists are updated and the new costs are determined for the remaining cells as given by Table 2.8. In the second iteration OR2 is selected and its input constraint - 1X0 is satisfied which finally defines the primary inputs as 110.

2.4 Estimation of Total Leakage Current Distribution

The leakage current distribution of a standard cell in state $i$ can be represented by a log-normal distribution with mean $\mu_i$ and standard deviation $D_i$. This distribution has a corresponding normal distribution having
Algorithm 2.1: Algorithm HeuristicMax

Algorithm HeuristicMax( Circuit Graph )

Input: CMOS circuit represented by a graph with cells as nodes and nets as edges

Output: Primary input vector for maximum leakage in the presence of process variations

Variables: Cells[cells], Nets[nets], Inputs[PI]

forall Nets do
  Sort in increasing order of depth from the primary inputs;
end
for i=1 to nets do
  Determine CC0 and CC1 lists;
end
for i=1 to cells do
  Generate PWICs;
  Compute PWLAs using table lookup;
end
for i=1 to cells do
  Generate conflicting cell list;
  Generate dominated cell list;
end
Make all primary inputs undefined;
Create a cell list with all cells;
while (cell list is not empty) and (primary inputs undefined) do
  Compute cost functions;
  Assign a large negative value to the cost of cells with infeasible inputs;
  Satisfy the input constraint of the cell with the highest cost;
  Remove the selected cell and its dominated cells from the cell list;
  Remove the conflicting cells from the cell list and push them to a violated cell list;
  Update conflicting and dominated cell lists for the remaining cells in the cell list;
end
for i=1 to PI do
  if Inputs(i) = X then
    Determine Leakage(1) = Total leakage contributed by cells with unsatisfied PWIC when Inputs(i)=1;
    Determine Leakage(0) = Total leakage contributed by cells with unsatisfied PWIC when Inputs(i)=0;
    if Leakage(1) > Leakage(0) then
      Inputs(i) = 1;
    else
      Inputs(i) = 0;
    end
  end
end
Table 2.8: Updated Values

<table>
<thead>
<tr>
<th>Cell</th>
<th>Conflicting Cells</th>
<th>Dominated Cells</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV-1</td>
<td>AND3</td>
<td>-</td>
<td>-0.406µ</td>
</tr>
<tr>
<td>AND3</td>
<td>INV-1, OR</td>
<td>-</td>
<td>LARGE NEG</td>
</tr>
<tr>
<td>BUF</td>
<td>OR</td>
<td>-</td>
<td>-0.520µ</td>
</tr>
<tr>
<td>OR2</td>
<td>AND3, BUF</td>
<td>INV-1</td>
<td>0.114µ</td>
</tr>
</tbody>
</table>

mean $m_i$ and standard deviation $σ_i$ obtained by taking the natural logarithm of all the points in the log-normal distribution [22].

2.4.1 Sum of Log-normal Probability Distributions

A standard cell in state $i$ with Gaussian mean $m_i$ and Gaussian standard deviation $σ_i$, has probability density function represented by Equation 2.3.

$$f(x) = \frac{1}{xσN\sqrt{2Π}} e^{-\frac{(ln(x)-m_i)^2}{2σ_i^2}}$$  (2.3)

Given the probability density function of all the standard cells, the total leakage power distribution of a circuit can be determined by approximation. In theory, the sum of log-normals is not known to have a closed form. An approximation of the same can be made using the Fenton-Wilkinson’s method of estimating the sum of several log-normal distributions [22]. A need for this approximation comes into picture because of the absence of a characteristic function for a log-normal distribution. On the other hand, in the case of normal distributions, the sum of two distributions can easily be found by their convolution integral. Fourier transforms which is the characteristic equation of a Gaussian distribution are used in the convolution process. The sum of two log-normal distributions is also a log-normal distribution. Given $f(x)$ of all the standard cells in the circuit, the sum leakage distribution is given by equating the first two moments, $α1$ and $α2$. Equation 2.4 gives the relationship between $α1$, $α2$, $m_i$ and $σ_i$. The logarithmic mean $μ_i$, is given by the first central moment, $α1$ and the variance is given by the difference between the second central moment and the square of the first central moment $(α2 − α1^2)$. Equations 2.5 and 2.6 give the relationship between $μ_i$, $D_i$, $m_i$ and $σ_i$. The resultant mean of the sum log-normal distributions is given by the sum of log-normal means of individual distributions and the resultant variance is equal to the sum of the log-normal variances of individual distributions. The mean $μ$ and variance $D^2$ of the sum log-normal distribution are given by
Equations 2.7 and 2.8 respectively and the distribution function is given by Equation 2.9.

\[
\begin{align*}
\alpha_1 &= e^{m_i \sigma_i^2/2}, \quad \alpha_2 = e^{2m_i \sigma_i^2} \\
\mu_i &= \alpha_1 = e^{m_i \sigma_i^2/2} \\
D_i^2 &= \alpha_2 - \alpha_1^2 = e^{2m_i \sigma_i^2} (e^{\sigma_i^2} - 1) \\
\mu &= \mu_1 + \mu_2 + \ldots + \mu_i + \ldots + \mu_n \\
D^2 &= D_1^2 + D_2^2 + \ldots + D_i^2 + \ldots + D_n^2 \\
f(x) &= \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(ln(x) - m)^2}{2\sigma^2}}
\end{align*}
\] (2.4)

2.4.2 Overall Approach

1. Calculate the maximum leakage vector using the heuristic described in Section 2.3.
2. Set the primary inputs of the circuit to the maximum leakage vector.
3. Forward-propagate the primary inputs and define the input states of all the gates in the circuit.
4. Estimate the probabilistic leakage distribution corresponding to the maximum vector using the method described in Section 2.4.1

2.5 Results

The approach described in the previous sections was implemented using C++ and was tested on ISCAS-85 benchmark circuits using predictive models for 32nm technology [1]. The nominal values for the process parameters that were varied are given by Table 1.1. Typical variation data is shown in Table 1.2. VDD for
32nm technology is 0.9V. The cells were characterized using 500 Monte Carlo simulations in hspice.

2.5.1 Leakage Power Distribution Corresponding to the Maximum Leakage Vector in the Presence of Variations

This sub-section gives the results for the heuristic implemented to determine the leakage power distribution corresponding to the maximum leakage vector in the presence of process parameter variations. The results are given by Table 2.9. This method was compared against random vector testing using 100,000 random vectors except for C17 which has 5 primary inputs, was exhaustively tested with $2^5$ input vectors. The average error for mean leakage current and standard deviation was found to be 1.32% and 1.41% respectively.

In the presence of process parameter variations, the mean leakage was several times larger than the nominal analysis as expected. The comparison results are given in Table 2.9. Figure 2.3 shows the comparison of means using the two approaches. Figure 2.4 shows the comparison of standard deviations using the two approaches. The mean leakage power obtained using this method was on an average 3.4X greater than the leakage current obtained using the method in [46]. It was observed that when the top two worst leakage states were considered instead of one worst leakage state, the accuracy of the algorithm improved on an average by 3.2% for mean and 2.7% for standard deviation. Only the smaller benchmarks were verified against Monte Carlo simulations using 100 random vectors due to the large run-time of Monte Carlo simulations. The results are shown in Table 2.10. Figure 2.5 shows the comparison of results using the two approaches.
Figure 2.4: Comparison of Heuristic Standard Deviation with Verified Standard Deviation

Figure 2.5: Comparison of Heuristic Mean with Monte Carlo Mean
Figure 2.6: Comparison of Heuristic Mean with Leakage Current Obtained Using [46]

Figure 2.7: Comparison of Heuristic Pessimistic Maximum with Verified Maximum
Figure 2.8: Comparison of Heuristic Pessimistic Maximum with the Leakage Obtained Using [46]

Figure 2.9: Comparison of Heuristic Mean with Leakage Obtained Using the Vector from [46]
Table 2.9: Comparison of Mean and Standard Deviation Obtained Using the Heuristic (H) with Random Vector Testing (R)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>H (\mu) ((\mu)A)</th>
<th>R (\mu) ((\mu)A)</th>
<th>Error (%)</th>
<th>H D (10^{-5})</th>
<th>R D (10^{-5})</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>2.4</td>
<td>2.4</td>
<td>0.0</td>
<td>0.27</td>
<td>0.27</td>
<td>0.0</td>
</tr>
<tr>
<td>C432</td>
<td>65.8</td>
<td>70.5</td>
<td>6.6</td>
<td>1.28</td>
<td>1.38</td>
<td>7.2</td>
</tr>
<tr>
<td>C499</td>
<td>123.3</td>
<td>127.5</td>
<td>3.3</td>
<td>1.52</td>
<td>1.60</td>
<td>5.0</td>
</tr>
<tr>
<td>C880</td>
<td>144.6</td>
<td>154.7</td>
<td>6.5</td>
<td>1.77</td>
<td>1.85</td>
<td>4.5</td>
</tr>
<tr>
<td>C1908</td>
<td>320.4</td>
<td>325.9</td>
<td>1.7</td>
<td>3.01</td>
<td>3.01</td>
<td>0.0</td>
</tr>
<tr>
<td>C2670</td>
<td>530.2</td>
<td>526.0</td>
<td>-0.1</td>
<td>3.46</td>
<td>3.50</td>
<td>1.3</td>
</tr>
<tr>
<td>C3540</td>
<td>709.7</td>
<td>731.3</td>
<td>2.9</td>
<td>4.14</td>
<td>4.21</td>
<td>1.6</td>
</tr>
<tr>
<td>C5315</td>
<td>997.9</td>
<td>979.1</td>
<td>-1.9</td>
<td>4.87</td>
<td>4.79</td>
<td>-1.7</td>
</tr>
<tr>
<td>C6288</td>
<td>653.2</td>
<td>642.2</td>
<td>-1.7</td>
<td>2.84</td>
<td>2.82</td>
<td>-0.1</td>
</tr>
<tr>
<td>C7552</td>
<td>1451.9</td>
<td>1394.4</td>
<td>-4.1</td>
<td>6.02</td>
<td>5.80</td>
<td>-3.7</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-</td>
<td>1.32</td>
<td>-</td>
<td>-</td>
<td>1.41</td>
</tr>
</tbody>
</table>

Table 2.10: Comparison of Heuristic Mean with Monte Carlo Mean

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>H (\mu) ((\mu)A)</th>
<th>MC (\mu) ((\mu)A)</th>
<th>Error (%)</th>
<th>Runtime Saving(X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>2.4</td>
<td>2.4</td>
<td>0.0</td>
<td>30,965</td>
</tr>
<tr>
<td>C432</td>
<td>65.8</td>
<td>66.5</td>
<td>1.1</td>
<td>59,520</td>
</tr>
<tr>
<td>C499</td>
<td>123.3</td>
<td>125.2</td>
<td>1.5</td>
<td>195,818</td>
</tr>
<tr>
<td>C880</td>
<td>144.6</td>
<td>148.5</td>
<td>2.6</td>
<td>337,042</td>
</tr>
</tbody>
</table>

2.5.2 Pessimistic Approach

A pessimistic approach was also implemented in which the maximum sampled value from the Monte Carlo simulations was used to calculate the maximum bound on leakage rather than the mean and standard deviation of the leakage profile. For example, the maximum sampled values for an inverter and an OR2 gate are given by Table 2.12. Using these maximum leakage values in the form of table look-up, the algorithm was used to compute the pessimistic leakage bound. The results obtained using this approach are given by Table 2.13 and the comparison between the pessimistic approach and the algorithm in [46] is given by Table 2.14. Figure 2.7 shows the comparison of pessimistic max with the verification method. Figure 2.8 shows the comparison between the pessimistic approach and the algorithm in [46]. The leakage computed using the pessimistic approach is 101.5X greater than the nominal value on an average. The average error of this approach when compared to random vector testing is 0.5%.
Table 2.11: Comparison of Mean Obtained Using Heuristic Approach (H) with the Leakage Current Obtained Using [46]

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Gates</th>
<th>PIs</th>
<th>[46] (µA)</th>
<th>H (µA)</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>6</td>
<td>5</td>
<td>0.57</td>
<td>2.4</td>
<td>4.20X</td>
</tr>
<tr>
<td>C432</td>
<td>160</td>
<td>36</td>
<td>19.64</td>
<td>65.8</td>
<td>3.35X</td>
</tr>
<tr>
<td>C499</td>
<td>202</td>
<td>41</td>
<td>38.55</td>
<td>123.3</td>
<td>3.19X</td>
</tr>
<tr>
<td>C880</td>
<td>383</td>
<td>60</td>
<td>44.43</td>
<td>144.6</td>
<td>3.25X</td>
</tr>
<tr>
<td>C1908</td>
<td>880</td>
<td>33</td>
<td>97.60</td>
<td>320.4</td>
<td>3.28X</td>
</tr>
<tr>
<td>C2670</td>
<td>1193</td>
<td>233</td>
<td>163.96</td>
<td>530.2</td>
<td>3.23X</td>
</tr>
<tr>
<td>C3540</td>
<td>1669</td>
<td>50</td>
<td>215.07</td>
<td>709.7</td>
<td>3.30X</td>
</tr>
<tr>
<td>C5315</td>
<td>2307</td>
<td>178</td>
<td>297.79</td>
<td>997.9</td>
<td>3.35X</td>
</tr>
<tr>
<td>C6288</td>
<td>2416</td>
<td>32</td>
<td>225.36</td>
<td>653.2</td>
<td>2.90X</td>
</tr>
<tr>
<td>C7552</td>
<td>3512</td>
<td>207</td>
<td>431.21</td>
<td>1451.9</td>
<td>3.37X</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.4X</td>
</tr>
</tbody>
</table>

Table 2.12: Maximum Sampled Leakage of Standard Cells

<table>
<thead>
<tr>
<th>Cell</th>
<th>Nominal Value (10^-7)</th>
<th>Max (10^-7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV (Input ‘0’)</td>
<td>1.49</td>
<td>59.9</td>
</tr>
<tr>
<td>INV (Input ‘1’)</td>
<td>0.82</td>
<td>133.6</td>
</tr>
<tr>
<td>OR2 (Input ‘00’)</td>
<td>2.29</td>
<td>163.40</td>
</tr>
<tr>
<td>OR2 (Input ‘01’)</td>
<td>1.53</td>
<td>725.50</td>
</tr>
<tr>
<td>OR2 (Input ‘10’)</td>
<td>0.92</td>
<td>68.58</td>
</tr>
<tr>
<td>OR2 (Input ‘11’)</td>
<td>0.73</td>
<td>68.39</td>
</tr>
</tbody>
</table>

2.5.3 Maximum Leakage Vector in the Presence of Variations

The maximum input vector obtained using the approach in [46] was used to calculate the leakage of the benchmark circuits in the presence of parameter variations. This leakage current was compared with the leakage current obtained using the vector computed by the developed heuristic which considers the effect of variations. The comparison results are given by Table 2.15. The average error in the mean leakage current was found to be 7.7% when parameter variations were not considered for the determination of maximum leakage vector.

2.5.4 Complexity and Usage

The implemented heuristic is quadratic in complexity. Leakage estimation using exhaustive hspice Monte Carlo simulations to determine the maximum leakage vector and the leakage power associated with it takes
a few hours to days depending on the number of random vectors considered and the size of the benchmarks.

The method developed in this work, estimates the same with less than 1.5% error in a matter of few seconds or a few minutes depending on the size of the benchmark. The runtime savings when compared to Monte Carlo simulations with a sweep value of 500 for 100 random vectors is given by Table 2.10. The CPU runtime of the overall approach is given by Table 2.15.

HeuristicMax accepts an RTL description of the CMOS circuit. It is assumed that all the cells in the standard cell library have been pre-characterized for leakage mean and standard deviation using Monte Carlo hspice simulations and are available to HeuristicMax in a look up table. The pre-characterization is a one time effort. The value of $\lambda$ is specified by the user. Using these inputs, HeuristicMax determines the maximum leakage vector and the leakage power associated with it. It can easily be used to accurately estimate the maximum leakage power of large circuits because of its small runtime when compared to the large run-time of Monte Carlo simulations.

### 2.5.5 Determination of Minimum Leakage Power Vector

HeuristicMax can be modified to determine the minimum leakage vector in the presence of process variations. In this modified approach, a probabilistic best input condition (PBIC) is defined. PBIC puts the standard cell into its least leakage state. The penalty for a cell for not settling into its PBIC, is defined as probabilistic cell leakage penalty (PCLP), given by the difference in the average of the high leakage states.
Table 2.14: Comparison of Leakage Obtained by the Pessimistic Approach (P) with Leakage Obtained Using [46]

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Gates</th>
<th># PIs</th>
<th>[46] (µA)</th>
<th>P (µA)</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>6</td>
<td>5</td>
<td>0.569</td>
<td>64.7</td>
<td>113.7X</td>
</tr>
<tr>
<td>C432</td>
<td>160</td>
<td>36</td>
<td>19.643</td>
<td>1742.8</td>
<td>88.7X</td>
</tr>
<tr>
<td>C499</td>
<td>202</td>
<td>41</td>
<td>38.554</td>
<td>4284.7</td>
<td>111.1X</td>
</tr>
<tr>
<td>C880</td>
<td>383</td>
<td>60</td>
<td>44.437</td>
<td>4380.0</td>
<td>98.6X</td>
</tr>
<tr>
<td>C1908</td>
<td>880</td>
<td>33</td>
<td>97.607</td>
<td>10114.3</td>
<td>103.6X</td>
</tr>
<tr>
<td>C2670</td>
<td>1193</td>
<td>233</td>
<td>163.965</td>
<td>17568.0</td>
<td>107.2X</td>
</tr>
<tr>
<td>C3540</td>
<td>1669</td>
<td>50</td>
<td>215.071</td>
<td>22387.1</td>
<td>104.1X</td>
</tr>
<tr>
<td>C5315</td>
<td>2307</td>
<td>178</td>
<td>297.791</td>
<td>31378.8</td>
<td>105.4X</td>
</tr>
<tr>
<td>C6288</td>
<td>2416</td>
<td>32</td>
<td>225.360</td>
<td>15915.1</td>
<td>70.7X</td>
</tr>
<tr>
<td>C7552</td>
<td>3512</td>
<td>207</td>
<td>431.215</td>
<td>46334.1</td>
<td>107.4X</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>101.5X</td>
</tr>
</tbody>
</table>

and the average of the low leakage states. Conflicting and dominated cells are defined in a similar way as HeuristicMax, but with reference to the PBIC. The calculation of costs is also similar to HeuristicMax but involves the usage of PCLP instead of the PWLA. The modified heuristic to determine the minimum leakage vector tries to minimize the cost penalty and hence finds the leakage vector corresponding to the minimum leakage power. The minimum leakage vector thus computed can be used to launch the circuit into a stand-by mode for leakage power minimization.
Table 2.15: Comparison of Leakage Means Using the Max Vector Obtained by Heuristic Approach (H) with the Max Vector Obtained by [46]

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>H $\mu$ ($\mu$A)</th>
<th>[46] $\mu$ ($\mu$A)</th>
<th>(%) Improvement</th>
<th>Runtime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>2.40</td>
<td>2.20</td>
<td>9.1</td>
<td>0.44</td>
</tr>
<tr>
<td>C432</td>
<td>65.80</td>
<td>63.75</td>
<td>3.2</td>
<td>8.37</td>
</tr>
<tr>
<td>C499</td>
<td>123.33</td>
<td>114.69</td>
<td>7.5</td>
<td>14.29</td>
</tr>
<tr>
<td>C880</td>
<td>144.57</td>
<td>133.72</td>
<td>8.1</td>
<td>59.94</td>
</tr>
<tr>
<td>C1908</td>
<td>320.40</td>
<td>297.16</td>
<td>7.8</td>
<td>184.54</td>
</tr>
<tr>
<td>C2670</td>
<td>530.23</td>
<td>498.42</td>
<td>6.4</td>
<td>2363.53</td>
</tr>
<tr>
<td>C3540</td>
<td>709.73</td>
<td>641.03</td>
<td>10.7</td>
<td>983.59</td>
</tr>
<tr>
<td>C5315</td>
<td>997.96</td>
<td>923.39</td>
<td>8.1</td>
<td>6022.29</td>
</tr>
<tr>
<td>C6288</td>
<td>653.28</td>
<td>601.64</td>
<td>8.6</td>
<td>1522.49</td>
</tr>
<tr>
<td>C7552</td>
<td>1451.9</td>
<td>1341.3</td>
<td>8.3</td>
<td>16283.24</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-</td>
<td>7.77</td>
<td>-</td>
</tr>
</tbody>
</table>
Chapter 3

Average Runtime Leakage Power Estimation with Process Variations

Runtime leakage power is the average leakage power dissipation of a circuit during its normal operation [37]. With leakage power contributing a significant amount to the total power dissipation, it becomes important to estimate the average leakage of a circuit during its normal operation. In the case of an application specific integrated circuit for instance, depending on its application, the chip has certain pre-determined / predicted signal probabilities for its primary inputs. As the signal probabilities traverse down the circuit graph, they become skewed and also become dependent and correlated due to re-convergent fan-outs. In such a scenario, it becomes important to consider the correlated input signal probabilities and compute the average leakage power dissipation of a circuit. The second method of estimation presented in this thesis work, aims to determine the run-time average leakage power dissipation in the presence of process parameter variations considering correlated signal probabilities.

3.1 Dependence of Average Leakage Power on Input Signal Probabilities

Consider a two input NOR gate. The leakage power mean and standard deviation of this gate in each of its input states is given by Table 3.1. The cell was characterized using 500 MC simulations in 32nm technology using ASU predictive models and variability data given by Table 1.2. It is observed that the mean leakage power in input state ‘00’ is almost 184.5X greater than the mean leakage power in input state ‘11’ which is the lowest leakage state. The standard deviation on the other hand in input state ‘00’ is 297.5X times
greater than the standard deviation in input state ‘11’. Due to such high disparity between leakage power of different states, it is important to consider the probability of occurrence of each input state during the normal operation of the circuit in order to derive an accurate estimate of the average leakage power dissipation. Lee et. al. in [36] have proposed an estimation technique to compute the average leakage power dissipation of a circuit by making use of signal probabilities, but this work does not take into consideration the effects of signal correlations due to re-convergent fan-out nodes and the effects of variability in process parameters. The work presented in this chapter addresses both these factors and attempts to accurately estimate the average run-time leakage power. It is assumed in this work that the signal probabilities of the primary inputs are known in advance depending on the application of the circuit.

### 3.2 Average Leakage Estimation and Optimization Flow

Figure 3.1 shows the overall flow for average leakage power estimation and optimization. The circuit netlist and input signal probabilities are fed to the leakage estimation heuristic. The output of the estimation heuristic is the mean and the standard deviation of the run-time average leakage power considering the effects of process variations. The leakage estimate derived in this manner is verified against an average value of a near exhaustive simulation using pseudo-random vectors. The verified leakage estimate is minimized using high threshold voltage devices with timing as a constraint. The optimization heuristic is described in chapter 4. The final output of this flow is a design optimized for leakage power.

### 3.3 Significance of Considering Run-time Signal Probabilities

Equal state probabilities cannot be assumed for all the nodes in the circuit for computing the average leakage power. As demonstrated in [36], the node state probabilities follow a bi-modal distribution. This means

<table>
<thead>
<tr>
<th>Input State</th>
<th>Leakage Mean (10-7)</th>
<th>Std. Dev. (10-7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>3.69</td>
<td>5.95</td>
</tr>
<tr>
<td>01</td>
<td>2.68</td>
<td>5.04</td>
</tr>
<tr>
<td>10</td>
<td>0.53</td>
<td>1.88</td>
</tr>
<tr>
<td>11</td>
<td>0.02</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Figure 3.1: Average Leakage Estimation and Optimization
that some nodes have high state probabilities and the others low state probabilities. Therefore, in order to accurately estimate the average leakage current, the assumption that all the nodes in a circuit have equal state probabilities is incorrect. As the signal probabilities, propagate through the circuit, they tend to diverge to high or low values as shown in [36]. This is illustrated by Figure 3.2 [36]. The example benchmark considered is MCNC benchmark circuit i10. The primary input state probabilities are assumed to be equally likely i.e. 0.5. It is however seen that the the state probabilities of the internal nodes and outputs follow a bi-modal distribution. The results of Figure 3.3 were obtained when the authors of [36] assumed the state probabilities of primary inputs to be either 0.2 or 0.8.
Table 3.2: Zero Algorithm

<table>
<thead>
<tr>
<th>Gate</th>
<th>‘1’ Signal Probability of output node Z [p_1(z)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>1 - ( p(a) )</td>
</tr>
<tr>
<td>AND</td>
<td>( p(a)p(b) )</td>
</tr>
<tr>
<td>OR</td>
<td>( p(a) + p(b) - p(a)p(b) )</td>
</tr>
<tr>
<td>XOR</td>
<td>( p(a) + p(b) - 2p(a)p(b) )</td>
</tr>
</tbody>
</table>

### 3.4 Computation of Node Signal Probabilities Considering Correlations

The computation of exact node signal probabilities is a #P-complete problem [25],[35]. Therefore, a heuristic with reduced complexity can be used to approximate the node signal probabilities. In this estimation approach, dynamic weighted averaging algorithm [21] is used, as it computes the node probabilities considering correlations with sufficient accuracy in less time.

The dynamic weighted averaging algorithm estimates the node signal probabilities by improving the weighted averaging algorithm [35]. The weighted averaging algorithm works only on re-convergent fanout nodes at the primary inputs by setting them to 0 and 1 selectively and hence does not account for the effects of the reconvergent fan-out nodes deep inside the circuit graph. The dynamic weighted averaging algorithm on the other hand, considers the effects of all the reconvergent fanout nodes in a circuit and hence is more accurate. It makes use of both the zero-algorithm and a modified version of the weighted averaging algorithm to achieve this.

#### 3.4.1 Zero Algorithm

The zero algorithm does not consider the effects of reconvergent fanout nodes. It assumes a tree structure for the network and computes the probabilities of the output nodes purely based on cell functionalities, given by Table 3.2. ‘a’ and ‘b’ in Table 3.2 are input node names and ‘z’ is the output node. \( p_1(i) \) is the probability that node ‘i’ is in state ‘1’. \( p_0(i) \) is the probability that node ‘i’ is in state ‘0’ and is given by 1-\( p_1(i) \).

#### 3.4.2 Weighted Averaging Algorithm

The weighted averaging algorithm recognizes the fact that reconvergent fanout nodes at the primary inputs leads to dependencies between the signals deeper in the circuit graph. This problem is addressed by forcing the reconvergent fanout node to ‘0’ and then ‘1’ selectively and hence computing the signal probabilities
given by Equation 3.1. Similarly, the ‘0’ signal probabilities are computed using Equation 3.2.

\[ p_{r1}(i) = p_1(i/r = 0)p(r = 0) + p_1(i/r = 1)p(r = 1) \]  
\[ p_{r0}(i) = p_0(i/r = 0)p(r = 0) + p_0(i/r = 1)p(r = 1) \]

where ‘i’ is the node whose signal probability is being determined and ‘r’ is the reconvergent fanout node affecting ‘i’.

### 3.4.3 Dynamic Weighted Averaging Algorithm (DWAA)

The dynamic weighted averaging algorithm extends the weighted averaging algorithm by applying the weighted averaging technique to all the reconvergent fanout nodes (RFONs) in the circuit including the ones which are internal to the circuit. The steps of the DWAA is given by Algorithm 3.1. The input to the algorithm is a CMOS circuit represented by a graph data structure. The standard cells in the circuit are represented by nodes in the graph and the nets are represented by edges. The algorithm also requires the state probabilities of the primary inputs of the circuit in the form of an input.

The first step in the algorithm is to sort all the nets in the circuit graph starting from the primary inputs. The re-convergent fan-out nodes are also sorted in increasing order of depth based on their position in the graph. Zero algorithm is used to compute the initial ‘0’ and ‘1’ signal probabilities for all nets in the circuit given by \( p_0(i, 0) \) and \( p_1(i, 0) \) respectively. The RFONs are processed in the order in which they appear in the sorted list and the signal probabilities of their fan-out cones are determined using the conditional probability equations given by Equations 3.1 and 3.2. The RFON is successively forced to ‘0’ and then to ‘1’ in order to compute the conditional probabilities of the fan-out cone. Finally, the weights of the signal probabilities are updated each time a node is processed using Equations 3.3 - 3.5 for \( P_1(i) \) and using Equations 3.6 - 3.8 for \( P_0(i) \). \( p_{r1}(input, t − 1) \) is the notation used for indicating that the ‘1’ signal probability of the \( i^{th} \) node is computed after processing (t-1) RFONs. Similarly, \( p_{r0}(input, t − 1) \) is the notation used for indicating that the ‘0’ signal probability of the \( i^{th} \) node is computed after processing (t-1) RFONs. The term fan-out cone represents the nets that are affected by the RFON considered.

The equation for updating \( p_1(i) \) is given by,
\[ p_1(i, t) = \frac{p_1(i, t-1)w_{s1}(i, t-1) + p_{r1}(i)w_{f1}(i)}{w_{s1}(i, t-1) + w_{f1}(i)} \] (3.3)

where the weighting factor is given by,

\[ w_{f1}(i) = \text{abs}(p_{r1}(i) - p_1(i, 0)) \] (3.4)

Equation 3.4 denotes the deviation on \( p_1(i) \) due to the effect of considering the RFON ‘r’.

\[ w_{s1}(i, t-1) = \sum_{k=1}^{t-1} w_{k1}(j) \] (3.5)

Equation 3.5 represents the sum of the weighting factors on the (t-1) processed RFONs so far w.r.t. to the computation of ‘1’ signal probability.

Similarly, the equation for updating \( p_0(i) \) is given by,

\[ p_0(i, t) = \frac{p_0(i, t-1)w_{s0}(i, t-1) + p_{r0}(i)w_{f0}(i)}{w_{s0}(i, t-1) + w_{f0}(i)} \] (3.6)

where the weighting factor is given by,

\[ w_{f0}(i) = \text{abs}(p_{r0}(i) - p_0(i, 0)) \] (3.7)

Equation 3.7 denotes the deviation on \( p_0(i) \) due to the effect of considering the RFON ‘r’.

\[ w_{s0}(i, t-1) = \sum_{k=1}^{t-1} w_{k0}(j) \] (3.8)

Equation 3.8 represents the sum of the weighting factors on the (t-1) processed RFONs so far w.r.t. to the computation of ‘0’ signal probability.

The DWAA is used to compute the ‘0’ and ‘1’ signal probabilities for all nets separately as in the case of correlated and dependent inputs, \( P_0(i) \) is not equal to \( 1 - P_1(i) \).
Algorithm 3.1: Algorithm DWAA

Algorithm DWAA( Circuit Graph, Signal Probabilities of PIs )

Input: CMOS circuit graph with cells as nodes and nets as edges
       Signal probabilities of the primary inputs

Output: Signal probabilities for all nets in the circuit

Variables: Cells[cells], Nets[nets], RFON[rfons], p(i,t)[nets]

forall Nets do
    Sort in increasing order of depth from the primary inputs;
    Determine the order of processing the RFONs starting from the lowest level;
end

for i=1 to nets do
    Compute $p_1(i, 0)$ using zero algorithm;
    Compute $p_0(i, 0)$ using zero algorithm;
end

for i=1 to rfons do
    Assign 0 and 1 successively to the RFON;
    Compute $p_{r1}(input, t - 1)$ of its fanout cone;
    Update $p_1(i)$ using the weighted averaging technique;
    Compute $p_{r0}(input, t - 1)$ of its fanout cone;
    Update $p_0(i)$ using the weighted averaging technique;
end
3.4.4 Illustration of Dynamic Weighted Averaging Algorithm

Consider the example circuit shown in Figure 3.4. The following steps summarize the application of DWAA on this combinational circuit to compute $P_1(i)$. The primary inputs are given by $i_1$, $i_2$, $i_3$ and $i_4$. The primary output is given by $o_1$. The internal nodes are given by $n_1$, $n_2$, $n_3$ and $n_4$. This example circuit has two reconvergent fan-out nodes $i_2$ at the primary input and $n_2$ which is an internal node. The fan-out cone of $i_2$ consists of nodes $n_1$, $n_2$, $n_3$, $n_4$ and $o_1$. The fanout cone of $n_2$ consists of $n_3$, $n_4$ and $o_1$. The signal probabilities for the primary inputs of this circuit are assumed to be 0.5. The internal node signal probabilities taking correlations into account for the RFONs at the primary inputs and also other RFONs internal to the circuit graph, are calculated by applying the DWAA as follows.

**Step 1. Zero Algorithm**: The signal probabilities of the primary inputs are set to 0.5. The zero algorithm is applied on this circuit using the equations given by Table 3.2. The signal probabilities obtained after the application of zero algorithm are given by Table 3.3.

**Step 2. Conditional Signal Probability**: The node signal probabilities computed using zero-algorithm are used by the conditional probability equation given by Equation 3.1 to obtain the correlated signal probabilities.

<table>
<thead>
<tr>
<th></th>
<th>$i_1$</th>
<th>$i_2$</th>
<th>$i_3$</th>
<th>$i_4$</th>
<th>$n_1$</th>
<th>$n_2$</th>
<th>$n_3$</th>
<th>$n_4$</th>
<th>$o_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumed</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Calculated</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.75</td>
<td>0.75</td>
<td>0.5625</td>
<td>0.875</td>
<td>0.49212</td>
</tr>
</tbody>
</table>
Table 3.4: Example - Dynamic Weighted Averaging Algorithm Results

<table>
<thead>
<tr>
<th></th>
<th>i1</th>
<th>i2</th>
<th>i3</th>
<th>i4</th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
<th>n4</th>
<th>o1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumed</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Calculated</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.75</td>
<td>0.75</td>
<td>0.625</td>
<td>0.875</td>
</tr>
</tbody>
</table>

Table 3.5: Example - Dynamic Weighted Averaging Algorithm Results

<table>
<thead>
<tr>
<th></th>
<th>i1</th>
<th>i2</th>
<th>i3</th>
<th>i4</th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
<th>n4</th>
<th>o1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumed</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.75</td>
<td>0.75</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Calculated</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.625</td>
<td>0.875</td>
<td>0.5809</td>
</tr>
</tbody>
</table>

**Step 3. Dynamic Weighted Averaging Technique**: Equations 3.3 to 3.5 are used to dynamically update the signal probabilities in the fan-out cone of the RFONs. The re-convergent fan-out node ‘i2’ is processed first followed by reconvergent fan-out node ‘n2’. The calculated $p_1(i)$ values after processing node ‘i2’ are given by Table 3.4. Finally, node ‘n2’ is processed and the updated values are given by Table 3.5. The final correlated signal probabilities for all nodes in the circuit is given by Table 3.6. Similarly, $p_0(i)$ can also be computed using the DWAA using the appropriate equations for the determination of ‘0’ signal probabilities.

### 3.5 State Probability Based Average Leakage Power

Consider a AND2 gate with leakage mean and standard deviation given by Table 3.9. This standard cell has inputs ‘A’ and ‘B’ and output ‘Z’. Assuming that $P_1(A)$ is 0.9 and $P_1(B)$ is 0.9, the input state probabilities for this particular gate is given by Table 3.7. However, if equal state probabilities were assumed, the results would have been drastically different as seen from Table 3.8. Based on this rationale [36], the state probability based mean leakage of a standard cell can be represented by Equations 3.9 and 3.10. Similarly, the state probability based standard deviation of a standard cell can be represented by Equations 3.11 and 3.12.

$$
\mu_i = \sum_{k=0}^{2^n} P_k \mu_k 
$$

(3.9)

where, $P_k$ is the probability of state-k, and $\mu_k$ is the mean leakage of gate-i in state-k given by,
Table 3.6: Example - Final Node Signal Probabilities

<table>
<thead>
<tr>
<th>i1</th>
<th>i2</th>
<th>i3</th>
<th>i4</th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
<th>n4</th>
<th>o1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.75</td>
<td>0.75</td>
<td>0.625</td>
<td>0.875</td>
<td>0.5809</td>
</tr>
</tbody>
</table>

Table 3.7: Input State Probabilities for AND2 Gate (\(P_1(A) = 0.9 \) and \(P_1(B) = 0.9\))

<table>
<thead>
<tr>
<th>Input State</th>
<th>State Probability</th>
<th>Leakage Mean (10^{-7})</th>
<th>State Probability * Leakage Mean (10^{-7})</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0.01</td>
<td>4.29</td>
<td>0.0429</td>
</tr>
<tr>
<td>01</td>
<td>0.09</td>
<td>6.16</td>
<td>0.5544</td>
</tr>
<tr>
<td>10</td>
<td>0.09</td>
<td>4.73</td>
<td>0.4257</td>
</tr>
<tr>
<td>11</td>
<td>0.81</td>
<td>8.64</td>
<td>6.9984</td>
</tr>
<tr>
<td>Resultant Mean</td>
<td>-</td>
<td>-</td>
<td>8.214</td>
</tr>
</tbody>
</table>

\[
\mu_k = e^{m_k} e^{\sigma_k^2/2} \quad \text{(3.10)}
\]

\[
D_i^2 = \sum_{k=0}^{k=2^n} P_k D_k^2 \quad \text{(3.11)}
\]

where, \(D_k^2\) is the leakage variance of gate-i in state-k given by,

\[
D_k^2 = e^{2m_k} e^{\sigma_k^2} (e^{\sigma_k^2} - 1) \quad \text{(3.12)}
\]

### 3.6 Sum of Log-normal Distributions

To re-iterate Fenton-Wilkinson method of computing the sum of several log-normal distributions from Chapter 2, the resultant mean of the sum log-normal distributions is given by the sum of log-normal means of individual distributions and the resultant variance is equal to the sum of the log-normal variances of individual distributions. The mean \(\mu\) and variance \(D^2\) of the sum log-normal distribution are given by Equations 3.13 and 3.14 and the distribution function is given by Equation 3.15.

\[
\mu = \mu_1 + \mu_2 + ... + \mu_i + ... + \mu_n \quad \text{(3.13)}
\]
Table 3.8: Input State Probabilities for AND2 Gate ($P_1(A) = 0.5$ and $P_1(B) = 0.5$)

<table>
<thead>
<tr>
<th>Input State</th>
<th>State Probability</th>
<th>Leakage Mean (10^{-7})</th>
<th>State Probability * Leakage Mean (10^{-7})</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0.25</td>
<td>4.29</td>
<td>1.07</td>
</tr>
<tr>
<td>01</td>
<td>0.25</td>
<td>6.16</td>
<td>1.54</td>
</tr>
<tr>
<td>10</td>
<td>0.25</td>
<td>4.73</td>
<td>1.18</td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td>8.64</td>
<td>2.16</td>
</tr>
<tr>
<td>Resultant Mean</td>
<td>-</td>
<td>-</td>
<td>5.95</td>
</tr>
</tbody>
</table>

where, $\mu$ is the log-normal mean of the sum distribution, and $\mu_i$ is the log-normal mean of the component distribution.

$$D^2 = D_1^2 + D_2^2 + ... + D_i^2 + ... + D_n^2$$ \hspace{1cm} (3.14)

where, $D^2$ is the log-normal variance of the sum distribution, and $D_i^2$ is the log-normal variance of the component distribution.

The final sum distribution function is given by,

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{-(\ln(x) - m)^2}{-2\sigma^2}}$$ \hspace{1cm} (3.15)

### 3.7 Overall Approach to Estimate the Average Leakage Power

1. Calculate the ‘0’ and ‘1’ signal probabilities for all the nets in the circuit using Algorithm 3.1.
2. For all gates, compute the mean leakage and standard deviation using Equations 3.9 - 3.12.
3. Compute the sum mean and variance for the circuit using Fenton Wilkinson’s method.

#### 3.7.1 Illustration of Overall Approach

The means and standard deviations for all input states of the standard cells used in the example circuit are given by Tables 3.9 - 3.10. The effective average leakage power associated with cell-1 (OR2) gate in the example circuit is given by Table 3.11 by applying Equation 3.9. The average standard deviation is obtained using Equation 3.11 and is also given by Table 3.11.
Table 3.9: Leakage Mean and Standard Deviation for AND2 Gate

<table>
<thead>
<tr>
<th>Input State</th>
<th>Mean (10^{-7})</th>
<th>Std. Dev. (10^{-7})</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>4.29</td>
<td>13.09</td>
</tr>
<tr>
<td>01</td>
<td>6.16</td>
<td>13.74</td>
</tr>
<tr>
<td>10</td>
<td>4.73</td>
<td>13.13</td>
</tr>
<tr>
<td>11</td>
<td>8.64</td>
<td>14.49</td>
</tr>
</tbody>
</table>

Table 3.10: Leakage Mean and Standard Deviation for OR2 Gate

<table>
<thead>
<tr>
<th>Input State</th>
<th>Mean (10^{-7})</th>
<th>Std. Dev. (10^{-7})</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>7.79</td>
<td>14.41</td>
</tr>
<tr>
<td>01</td>
<td>5.00</td>
<td>7.65</td>
</tr>
<tr>
<td>10</td>
<td>2.52</td>
<td>4.54</td>
</tr>
<tr>
<td>11</td>
<td>2.03</td>
<td>4.36</td>
</tr>
</tbody>
</table>

Resultant Leakage Mean of OR2 = 4.34e-7
Resultant Leakage Standard Deviation of OR2 = 7.73e-7

Similarly, the means and standard deviations for the average leakage power associated with every cell in the circuit are obtained. The mean and standard deviation for the average leakage power distribution of the example circuit are calculated using equations 3.13 and 3.14. The computed values for mean and standard deviation are given below.
Total Leakage Mean = 2.37e-06
Resultant Standard Deviation = 1.66e-06

3.8 Results

The overall approach was implemented in C++ which uses hspice Monte Carlo characterization results in the form of a table look-up. 500 MC simulations were performed for every cell in every input condition to characterize the standard cells. The implemented heuristic was tested on ISCAS-85 benchmark circuits and the results are given by Table 3.12. The state probabilities for the primary inputs were assumed to be 0.5. The results were verified using 100,000 random vectors to obtain the mean and standard deviation
Table 3.11: Leakage Mean and Standard Deviation of Cell-1 Considering Signal Probabilities

<table>
<thead>
<tr>
<th>Input State</th>
<th>$P_k$</th>
<th>Effective Mean = $P_k \times \text{Mean (10}^{-7})$</th>
<th>Effective Std. Dev. = $P_k \times \text{Std. Dev. (10}^{-7})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0.25</td>
<td>1.95</td>
<td>3.6</td>
</tr>
<tr>
<td>01</td>
<td>0.25</td>
<td>1.25</td>
<td>1.91</td>
</tr>
<tr>
<td>10</td>
<td>0.25</td>
<td>0.63</td>
<td>1.13</td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td>0.51</td>
<td>1.09</td>
</tr>
</tbody>
</table>

Table 3.12: Run-time Average Leakage Current Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>H Mean (10^{-6})</th>
<th>R Mean (10^{-6})</th>
<th>Error (%) Mean</th>
<th>H Std. (10^{-5})</th>
<th>R Std. (10^{-5})</th>
<th>Error (%) Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>1.84</td>
<td>1.85</td>
<td>0.68</td>
<td>0.23</td>
<td>0.23</td>
<td>0.60</td>
</tr>
<tr>
<td>C432</td>
<td>61.75</td>
<td>62.04</td>
<td>0.46</td>
<td>1.26</td>
<td>1.27</td>
<td>0.40</td>
</tr>
<tr>
<td>C499</td>
<td>118.89</td>
<td>118.93</td>
<td>0.02</td>
<td>1.42</td>
<td>1.42</td>
<td>0.04</td>
</tr>
<tr>
<td>C880</td>
<td>130.19</td>
<td>139.50</td>
<td>6.67</td>
<td>1.62</td>
<td>1.67</td>
<td>3.34</td>
</tr>
<tr>
<td>C1908</td>
<td>314.31</td>
<td>315.05</td>
<td>0.23</td>
<td>2.92</td>
<td>2.91</td>
<td>0.16</td>
</tr>
<tr>
<td>C2670</td>
<td>474.53</td>
<td>514.34</td>
<td>7.73</td>
<td>3.23</td>
<td>3.39</td>
<td>4.76</td>
</tr>
<tr>
<td>C3540</td>
<td>610.84</td>
<td>670.91</td>
<td>9.01</td>
<td>3.87</td>
<td>4.03</td>
<td>3.89</td>
</tr>
<tr>
<td>C6288</td>
<td>574.17</td>
<td>581.93</td>
<td>1.33</td>
<td>2.55</td>
<td>2.66</td>
<td>4.10</td>
</tr>
<tr>
<td>C7552</td>
<td>1287.79</td>
<td>1354.19</td>
<td>4.9</td>
<td>5.49</td>
<td>5.65</td>
<td>2.85</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-</td>
<td>3.45</td>
<td>-</td>
<td>-</td>
<td>2.24</td>
</tr>
</tbody>
</table>

The results of the heuristic were compared with the verification results. It is seen from the results that the error for mean leakage power was 3.45% and the error for standard deviation was 2.24%. Figure 3.5 shows the comparison between the heuristic mean and the mean obtained by averaging the random vector verification method. Figure 3.6 shows the comparison between the heuristic standard deviation and the standard deviation obtained from the verification method.
Figure 3.5: Comparison of Heuristic Mean with Verified Mean

Figure 3.6: Comparison of Heuristic Standard Deviation with Verified Standard Deviation
Chapter 4

Optimization of Average Runtime Leakage Power in the Presence of Process Variations

4.1 Optimization Techniques for Leakage Power Reduction

There are several techniques that can be adopted to minimize leakage power in CMOS circuits. Methods like input vector control, use of dual threshold voltage devices, channel length biasing, power supply gating and body biasing are widely used at the circuit level in order to reduce the leakage power consumption. Some of the important techniques used to minimize leakage power consumption are briefly described in the following section.

4.1.1 Input Vector Control

Input Vector Control makes use of the intrinsic stack effect [33] of the standard cells in order to launch the cell into its lowest leakage state. For example, in the case of a two input NAND gate, input ‘00’ maximizes the number of OFF transistors in the NMOS stack and hence launches it into its lowest leakage state. This technique does not require any circuit modifications or re-design. Use of exhaustive techniques to compute the vector that results in minimum leakage for a large circuit is prohibitive due to its exponential complexity. Several methods for implementing input vector control with polynomial time complexity have been proposed in the past [14],[32],[30],[18], [24],[41],[8],[27],[19]. Although this technique is effective, it can be used only in the stand-by mode and hence cannot be used to minimize the run-time leakage power.
dissipation.

4.1.2 Dual Threshold Voltage Technique

The leakage power dissipation in high $V_{th}$ devices is far less compared to the nominal $V_{th}$ / low $V_{th}$ devices. At the same time, the delay associated with high $V_{th}$ cells is higher compared to nominal $V_{th}$ cells. Due to this reason, the dual $V_{th}$ technique makes use of low / nominal threshold devices in the critical timing paths and high threshold devices in non-critical paths. Work in [53],[40],[55],[44],[50] propose methods to implement this technique in order to reduce leakage power.

4.1.3 Channel Length Biasing

Increase in channel length increases the threshold voltage and in turn exponentially reduces leakage power consumption [51]. But, on the other hand, increase in channel length linearly increases the delay and dynamic power. This technique was first proposed by [51] and was later improved in [28], [29].

4.1.4 Power Supply Gating

Power gating is implemented by the use of a high $V_{th}$ header (PMOS) or a footer (NMOS) switch which puts the circuit block into sleep mode when the circuit is idle for a long time. The sleep signal connected to the header / footer switch is used to de-active the circuit block. When the logic block is required again, the sleep signal is de-asserted in order to connect the block to the VDD / Ground rails. This technique helps to reduce the sub-threshold leakage drastically in the idle mode. Several work [54],[57],[31] propose methods to implement this technique in order to reduce leakage power.

4.1.5 Body Biasing

Reverse body biasing of MOSFET is used to reduce the sub-threshold leakage while forward body biasing is used to reduce the BTBT leakage current [34]. Since the two types of leakage need opposing types of bias for their reduction, it becomes necessary to find an optimum bias voltage for leakage minimization. This has been addressed in [43].
Table 4.1: Leakage Mean and Standard Deviation of 2 Input NAND

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Nominal $V_{th}$ Mean (e-09)</th>
<th>High $V_{th}$ Mean (e-09)</th>
<th>Nominal $V_{th}$ Std. (e-09)</th>
<th>High $V_{th}$ Std. (e-09)</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘00’</td>
<td>3.33</td>
<td>1.46</td>
<td>2.04</td>
<td>0.79</td>
</tr>
<tr>
<td>‘01’</td>
<td>44.84</td>
<td>11.21</td>
<td>94.17</td>
<td>17.40</td>
</tr>
<tr>
<td>‘10’</td>
<td>167.28</td>
<td>48.85</td>
<td>296.11</td>
<td>88.77</td>
</tr>
<tr>
<td>‘11’</td>
<td>788.46</td>
<td>192.56</td>
<td>1771.41</td>
<td>657.84</td>
</tr>
</tbody>
</table>

4.2 Dual Threshold Voltage Approach

In this work, the leakage that needs to be optimized is the mean and standard deviation of the run-time average leakage power. Since the target leakage is run-time leakage, techniques like power gating, body biasing, input vector control, etc. cannot be used as these techniques are primarily used in the stand-by or idle mode. Due to this reason, the minimization of average run-time leakage power in this work is done using the dual threshold voltage approach. This technique is used due to the huge disparity in leakage means and standard deviations between nominal and high $V_{th}$ devices. Table 4.1 shows the leakage means and standard deviations of a two input NAND gate for both nominal and high $V_{th}$ cells derived by performing 500 Monte Carlo simulations. From this table, it is seen that the leakage mean and standard deviation of nominal $V_{th}$ NAND in input state ‘11’ is 4.1X higher than high $V_{th}$ NAND. The standard deviation of nominal $V_{th}$ NAND in input state ‘11’ is 2.7X higher than high $V_{th}$ NAND gate. Both the leakage mean and standard deviation can be minimized using the dual $V_{th}$ approach and the characterized data of standard cells predict a huge reduction in leakage power. Therefore, this technique is used for leakage optimization.

4.3 Timing Constraints in Leakage Optimization Using Dual Threshold Cells

The use of high threshold cells for leakage minimization increases the delay penalty in the circuit. Therefore, it becomes necessary that the high threshold cells are used only in non-critical timing paths. This necessitates the timing characterization of the standard cells in the technology to obtain the timing information of the circuit. The delay values of a nominal and high $V_{th}$ NAND gate in 32nm technology is given by Table 4.2. The standard cells were characterized using Signal Storm and were used in the form of a table look-up to compute the timing information in the optimization algorithm.
Table 4.2: Delay Values for 2 Input NAND

<table>
<thead>
<tr>
<th>Fanout</th>
<th>Nominal $V_{th}$ Delay (e-11)</th>
<th>High $V_{th}$ Delay (e-11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.44</td>
<td>7.48</td>
</tr>
<tr>
<td>1</td>
<td>5.21</td>
<td>8.33</td>
</tr>
<tr>
<td>2</td>
<td>5.96</td>
<td>9.41</td>
</tr>
<tr>
<td>3</td>
<td>6.65</td>
<td>10.36</td>
</tr>
<tr>
<td>4</td>
<td>7.26</td>
<td>11.24</td>
</tr>
</tbody>
</table>

### 4.4 Optimization Approach

The algorithm used to minimize the run-time average leakage power is described in this section. The deterministic approach proposed by Srivastava et. al. in [53] was modified to include the effects of parameter variations. The algorithm steps are given by Algorithm 4.1. The input to the heuristic is the circuit graph and a pre-determined timing penalty is assumed for sufficient leakage reduction. Algorithm 3.1 described in Chapter 3 is used to determine the average run-time leakage power. Once the leakage estimate is obtained, the timing information is obtained in the form of critical path delay of the circuit. The timing path with the maximum delay is defined as the critical path of the circuit. The path delays for all other paths in the circuit is also found in this step and the slack w.r.t. the critical path is computed for all paths. The equation for slack computation is given by Equation 4.1.

$$\text{Slack}_{path} = \text{Critical Path Delay} - \text{Path Delay} \quad (4.1)$$

For a cell that is present in multiple paths in the circuit, the minimum slack on the cell is assigned as the final slack of the cell. In the heuristic described in this section, a timing penalty of 10% is assumed. This means that the critical path delay can be increased by another 10% in the circuit while a sufficient leakage reduction can be obtained. In order to determine which cell needs to be replaced, the sensitivity equation from [53] was adapted to take process variations into consideration.

The sensitivity factor used in the algorithm is given by,

$$\text{Sensitivity} = \frac{\Delta P}{\Delta D} \cdot \text{Slack}_{path} \quad (4.2)$$

54
where,

$$\Delta P = \text{Leakage Power of Nominal } V_{th} \text{ Cell} - \text{Leakage Power of High } V_{th} \text{ Cell} \quad (4.3)$$

$$\Delta D = \text{Propagation Delay of High } V_{th} \text{ Cell} - \text{Propagation Delay of Nominal } V_{th} \text{ Cell} \quad (4.4)$$

$$\text{Leakage Power} = 0.5 \times \text{Leakage Mean} + 0.5 \times \text{Leakage Standard Deviation} \quad (4.5)$$

The sensitivity factor thus obtained, determines that the cell with the maximum sensitivity cost is more likely to yield maximum leakage reduction. The cell with maximum sensitivity is replaced with a high $V_{th}$ cell. This replacement changes the timing of all the paths / cells that are affected by this change. Hence, the timing of paths and the slack of cells affected by this change are updated accordingly. If the affected path is the critical path, then the critical path delay is also updated. The high $V_{th}$ cell replacement step is done iteratively till the assumed timing penalty is reached and this is the stopping criteria for the algorithm.

### 4.5 Results

The implemented heuristic was tested on ISCAS-85 benchmark circuits. The reduction obtained in leakage current mean and standard deviation is shown in Table 4.3. An average improvement of 44.01% was seen for mean leakage power and an average improvement of 43.72% was seen for standard deviation. Figure 4.1 shows the comparison of pre-optimized and post-optimized means. The comparison between pre-optimized standard deviation and post-optimized standard deviation is shown by Figure 4.2.
Algorithm 4.1: Algorithm LeakageOptimization

Algorithm LeakageOptimization( Circuit Graph, Timing Penalty )

Input:   CMOS circuit represented by a graph
         with cells as nodes and nets as edges
         Timing Penalty = 10%

Output:  Leakage optimized circuit

Variables: Cells[cells], Nets[nets]

Determine the run-time average leakage power using Algorithm - 3.1;
Determine the critical paths and the critical delay for the circuit;
forall Paths do
    Compute path delay;
    Compute slack w.r.t. the timing critical path;
end
CriticalTiming = Critical Path Delay;
while (CriticalTiming ≤ 1.1(Critical Path Delay)) do
    forall Cells do
        Compute Sensitivity;
    end
    Replace the cell with maximum sensitivity with a high $V_{th}$ cell;
    Update timing information for all the affected timing paths;
    Update slack for all affected cells;
    Update CriticalTiming;
    Update run-time leakage power;
end

Figure 4.1: Optimized Leakage Mean Results
Table 4.3: Optimization Algorithm Results (10% Timing Penalty)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>$H$ Mean($10^{-6}$)</th>
<th>Opt Mean($10^{-6}$)</th>
<th>Reduction(%)</th>
<th>$H$ Std.($10^{-5}$)</th>
<th>Opt Std.($10^{-5}$)</th>
<th>Reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>1.84</td>
<td>1.39</td>
<td>24.15</td>
<td>0.23</td>
<td>0.19</td>
<td>15.4</td>
</tr>
<tr>
<td>C432</td>
<td>61.75</td>
<td>46.93</td>
<td>23.99</td>
<td>1.27</td>
<td>1.00</td>
<td>20.6</td>
</tr>
<tr>
<td>C499</td>
<td>118.89</td>
<td>96.74</td>
<td>18.63</td>
<td>1.42</td>
<td>1.23</td>
<td>13.6</td>
</tr>
<tr>
<td>C880</td>
<td>130.19</td>
<td>44.44</td>
<td>65.86</td>
<td>1.62</td>
<td>0.72</td>
<td>55.5</td>
</tr>
<tr>
<td>C1908</td>
<td>314.31</td>
<td>125.15</td>
<td>60.17</td>
<td>2.92</td>
<td>1.50</td>
<td>48.6</td>
</tr>
<tr>
<td>C2670</td>
<td>474.53</td>
<td>157.78</td>
<td>66.75</td>
<td>3.23</td>
<td>1.28</td>
<td>60.23</td>
</tr>
<tr>
<td>C3540</td>
<td>610.84</td>
<td>193.46</td>
<td>68.32</td>
<td>3.87</td>
<td>1.51</td>
<td>60.93</td>
</tr>
<tr>
<td>C5315</td>
<td>779.29</td>
<td>245.99</td>
<td>68.43</td>
<td>4.24</td>
<td>1.71</td>
<td>59.6</td>
</tr>
<tr>
<td>C7552</td>
<td>1287.79</td>
<td>409.84</td>
<td>68.18</td>
<td>5.49</td>
<td>2.25</td>
<td>59.03</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-</td>
<td>44.01</td>
<td>-</td>
<td>-</td>
<td>43.72</td>
</tr>
</tbody>
</table>

Figure 4.2: Optimized Leakage Standard Deviation Results
Chapter 5

Conclusion

5.1 Objectives

In this research work, three important objectives were considered and met.

1. The estimation of maximum leakage power bound in the presence of process variations
2. The estimation of run-time average leakage power distribution in the presence of process variations
3. Optimization of average leakage power considering timing as a constraint

Chapter 1 in this work explains the background and previous work done in the area of statistical leakage power estimation and optimization. It also discusses the importance of process variability and its effects on important performance measures like leakage power and timing. The different types of defects seen with technology scaling are also explained briefly in this chapter. The dependence of leakage power, in particular on process parameters like channel length, gate width, threshold voltage, gate oxide thickness and channel doping concentration are explained in detail with experimental results.

5.2 Estimation of Maximum Leakage Power Bound

Chapter 2 discusses the method of estimation of statistical leakage power corresponding to the maximum leakage vector in the presence of process variations. It was shown in this work, that maximum leakage power is dependent not only on the input vectors but also on process variations. Since corner case analysis is pessimistic and nominal analysis is optimistic in nature, an accurate measurement is possible only through
statistical analysis. Such an accurate analysis is important in identifying important power and reliability issues which affect leakage power dissipation and electromigration.

The statistical approach used in this estimation technique determines the mean and the standard deviation of leakage power of ISCAS-85 benchmark circuits. It was shown in this work that the mean of the maximum leakage power distribution is 3.4X the nominal leakage power value. This clearly demonstrates that if variability data is not considered, the analysis in such a case will be inaccurate. Therefore, it becomes important to consider the effects of variations in process parameters especially channel length, threshold voltage and thickness of gate oxide due to the exponential dependence of leakage power on them. The results of this approach when verified against 100,000 random vectors gave an average error of 1.32% for mean leakage and 1.41% for standard deviation. There was an improvement of 7.7% compared to the deterministic approach in [46] when process variations were considered for the computation of maximum leakage vector.

A non-statistical approach was also developed to estimate the maximum leakage power bound based on the maximum sampled leakage values in the Monte Carlo simulations. The leakage power estimated using this approach was found to be on an average 101.05X times greater that the nominal leakage value. It was determined that this approach gave pessimistic results compared to the statistical approach. The error compared to the verification method that uses 100,000 random vectors was 0.5%.

5.3 Estimation of Average Leakage Power Distribution

Chapter 3 discusses the method developed to estimate the run-time average leakage power distribution. Considering random vectors for the computation of average leakage power is prohibitive due to the large sample size of the input vectors required. The heuristic developed to meet this objective considers signal probabilities for the primary input vector. It also considers the effects of correlations due to re-convergent fan-out nodes in the circuit graph. The verification method uses 100,000 random vectors to compute the average leakage power distribution over all the input vectors considered. The error when compared to this verification method was found to be 3.45% for mean leakage power and 2.24% for standard deviation.
5.4 Optimization of Average Leakage Power Distribution

Chapter 4 explains the various optimization techniques used for the optimization of leakage power. In this work, dual threshold voltage devices are used to optimize the average leakage power distribution. Since leakage power optimization using high threshold devices affects circuit timing, the heuristic optimizes leakage with timing as a constraint. A sensitivity cost associated with every cell is computed based on the leakage power savings, timing penalty and slack on the path in which the cell is present. The cell with the maximum sensitivity is swapped with a high threshold voltage cell to reduce the leakage power consumption. The experimental results consider leakage savings for a timing penalty of 10%. It was shown that an average savings of 44.01% for leakage mean and 43.72% for standard deviation was obtained using this method.

5.5 Future Work

In this research work, only random variations were considered for analysis. This work can be extended to include the effects of correlated process variations. Layout level analysis will further increase the accuracy of the approach by including the effect of spatial correlations. Layout level extraction will also yield parasitic capacitances which alter the delay and power values. Therefore, this work can be verified by doing a layout level analysis. Band to Band tunneling leakage and Gate Induced Drain Leakage are forms of leakage that are less dominant in the current technologies but are expected to increase in future technologies. This work can also be extended to include other leakage components to further increase the accuracy of this method. Environment variations such as temperature and voltage variations can also be included to extend this work.

In conclusion, leakage power is becoming a dominant factor in the total power consumption of a CMOS circuit and hence needs to be estimated accurately. Since the variations in process parameters have a huge impact on the leakage power consumption, it is important to consider their effects for leakage power estimation and optimization. These factors make it necessary to adopt statistical methods for leakage analysis and minimization. In this work, two techniques for the statistical estimation of leakage power dissipation and a leakage optimization technique were developed and implemented. They were further tested with ISCAS-85 benchmark circuits. The developed techniques were also verified and found to provide results with sufficient accuracy and reduced run-time compared to spice simulations.
Publication

Bibliography


Appendix A

Hspice Monte Carlo Experimental Setup

A.1 Monte Carlo Analysis in Hspice

The Monte Carlo analysis feature of hspice is used for simulating the effects of variations in process parameters or device characteristics on the performance of the circuit. The variations in device characteristics can be defined as a gaussian or a uniform distribution function. The specification also includes relative or absolute variation of a certain parameter with respect to a nominal value at the sigma level. Using this information, random values are chosen by the tool to create the input sample space which are used in successive simulations to determine the range of performance values. This can be termed as a distribution and can be represented by a mean value and a standard deviation. When fewer number of input samples are used, the distribution is not well defined. The increase in the number of samples well defines the shape of the distribution and makes it smoother. Though Monte Carlo simulation is referred to as the golden reference, it is impossible to use this analysis on large circuits because of the expensive computation time that is involved in this kind of analysis. Figure A.1 shows the Monte Carlo analysis flow [4] in hspice.

A.2 Monte Carlo Simulation Setup

The Monte Carlo simulation setup used in this work is described in this section.

1. The .option section of the spice deck includes the modmonte = 1 option to specify that Monte Carlo analysis is set to 1 and all the parameters that are varied receives a different random value.

2. The .param section includes all the parameters that are intended to vary in the simulation. The parameters
Figure A.1: Monte Carlo Analysis in Hspice [Source:[4]]
that were varied are channel length, gate width, thickness of gate oxide, channel doping concentration and threshold voltage. The distribution function used to vary the parameters is the relative gaussian distribution with a sigma value of 3.

3. The Monte Carlo analysis is always used along with another type of simulation like the DC, AC or transient simulation. The experimental setup in this work uses Monte Carlo simulation in conjunction with transient simulation. A start and stop time along with a Monte Carlo sweep value is described in this statement. A lower sweep value means less simulation time but the distribution is not well-defined. On the other hand, a high sweep value defines the distribution better but increases the simulation time. Therefore, an optimum number is chosen to trade-off between the two constraints. In this experimental setup, a sweep value of 500 is used.

Table A.1 summarizes the Monte Carlo simulation experimental setup with examples [3].

<table>
<thead>
<tr>
<th>Hspice Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.OPTION MODMONTE = 1</td>
<td>Introduces Random Variation</td>
</tr>
<tr>
<td>.PARAM lp = GAUSS(2, 0.12, 3)</td>
<td>Gaussian distribution function using relative variation</td>
</tr>
<tr>
<td>.TRAN 1n 10n SWEEP MONTE = 500</td>
<td>Sweep Value of Monte Carlo Simulation</td>
</tr>
</tbody>
</table>
Appendix B

Timing Library Experimental Setup

B.1 Signal Storm Library Characterization

Signal Storm from Cadence was used to characterize the timing library of standard cells. The delay values were computed for both nominal threshold and high threshold voltage devices. Signal Storm creates .lib or the Liberty format database which is the standard format for timing data.

The tool requires the following inputs in order to characterize the standard cells:

1. Supply voltage (Example : 0.9V)
2. Temperature (Example : 25)
3. Process corner (Example : Typical)
4. Input slew (Example : 10ps)
5. Output load (Example : 10fF)
6. Cell functionality (Example : !INPUT for Inverter cell)

Signal Storm creates the delay vectors automatically and uses them for simulating the standard cells. The output database includes the rise and fall delays and rise and fall transitions which can be used by a static timing analysis tool such as Prime Time for generating timing information.

B.2 Liberty Format

Following is the .lib format for an example inverter cell (INV1) characterized using Signal Storm.
/* ————- *
Design : INV1 *
————- */
cell (INV1) {
  area : 0.0;
cell_leakage_power : 16.4708;
  pin(INPUT) {
    direction : input;
    capacitance : 0.000464512;
    rise_capacitance : 0.000464512;
    fall_capacitance : 0.000442781;
    rise_capacitance_range ( 0.000464512, 0.000464512 ) ;
    fall_capacitance_range ( 0.000442781, 0.000442781 ) ;
  }
  pin(OUTPUT) {
    direction : output;
    function : “(!INPUT)”;
    timing() {
      related_pin : “INPUT”;
      timing_sense : negative_unate;
      cell_rise(delay_template_6x6) {
        index_1 (”0.1, 0.5, 1.2, 3, 4, 5”);
        index_2 (”0.06, 0.24, 0.48, 0.9, 1.2, 1.8”);
        values (”0.520832, 0.567176, 0.63622, 0.774692, 0.850953, 0.96875”,
          “2.62877, 2.60423, 2.63351, 2.82694, 2.88773, 3.03372”,
          “6.13721, 6.18011, 6.23319, 6.34853, 6.45937, 6.49341”,
          “15.1044, 15.1322, 15.1127, 15.2913, 15.3932, 15.3078”,
          “72
\begin{verbatim}
}

rise_transition(delay_template_6x6) {
    index_1 ("0.1, 0.5, 1.2, 3, 4, 5");
    index_2 ("0.06, 0.24, 0.48, 0.9, 1.2, 1.8");
    values ("0.816305, 0.828235, 0.946731, 0.945096, 1.06836, 1.36896",
            "4.07777, 4.10872, 4.10274, 4.14366, 4.13511, 4.23602",
            "32.8673, 32.767, 32.7319, 32.8732, 32.8823, 32.7818",
            "40.9895, 40.9988, 41.0311, 41.0116, 40.9403, 41.0575");
}

cell_fall(delay_template_6x6) {
    index_1 ("0.1, 0.5, 1.2, 3, 4, 5");
    index_2 ("0.06, 0.24, 0.48, 0.9, 1.2, 1.8");
    values ("0.406327, 0.463201, 0.517996, 0.645911, 0.698378, 0.764678",
            "2.13299, 2.08257, 2.11418, 2.1681, 2.26999, 2.40625",
            "4.76736, 5.09873, 5.08175, 4.91987, 5.22889, 5.22154",
            "11.6494, 11.8523, 11.9349, 11.8923, 12.0097, 12.2426",
            "15.6456, 15.6588, 15.6455, 15.8461, 15.8974, 15.9554",
}

fall_transition(delay_template_6x6) {
    index_1 ("0.1, 0.5, 1.2, 3, 4, 5");
    index_2 ("0.06, 0.24, 0.48, 0.9, 1.2, 1.8");
    values ("0.602151, 0.580462, 0.594859, 0.689043, 0.871923, 1.24606",
}
\end{verbatim}
“2.97172, 2.89388, 2.90614, 3.33373, 2.84553, 3.0881”,
“7.05811, 7.05245, 6.88585, 7.19413, 7.56783, 6.85145”,
“17.0161, 17.2315, 17.177, 16.9125, 17.2377, 17.0402”,
“22.0762, 22.4786, 22.561, 22.1662, 22.3853, 22.5336”,
“27.6663, 27.8927, 27.9362, 27.7315, 27.5763, 27.9924”);