

國立交通大學

電子工程學系電子研究所

博士論文

超大型積體電路功率估測及功率分佈繪圖之研究

On Power Consumption Estimation and

Power Distribution Profiling for VLSI Circuits

研究生：黃恆亮

指導教授：周景揚教授

中華民國九十三年六月

中文摘要

隨著半導體製程的進步，半導體線路中的金屬導線越來越細，導線與導線之間的距離也越來越近，這樣的現象使得散熱問題不得不為線路設計者所重視。更有甚者，積體電路的工作頻率也不斷的向上升高，同樣也使得線路中因功率消耗所轉換的熱能難以宣洩。線路溫度的提高最直接的結果就是晶片冒煙燒毀。但是在燒毀之前它也可能因為導線在高溫下長期使用而斷裂，或者是電阻值在高溫下增大到電路功能失常。這些現象都顯示了線路設計者在生產前做好功率估測的重要性。

然而功率估測的方式有很多種，比如，估計電池耐久時間時要計算能量消耗，分析溫度變化要測量功率消耗，評估導線寬度要測量平均電流，而導線可靠度分析則需要電流密度的平均值及方均根值等等。雖然這些測量值都有明確的定義及適用場合，但是卻都會遇到一個相同的問題就是：功率估測值是和輸入訊號有關的，但是線路設計者卻通常無法想像出一個具有代表性的輸入訊號來進行功率估測。

針對這個問題，本篇論文提出了一個結合了最準確的線路模擬-SPIICE 的全自動解決方法。首先本論文先分析了輸入訊號的統計值的特性以及這些特性和功率消耗之間的關係。我們證實了每個不同的輸入訊號應該用不同的功率靈敏度。而且我們發現這種功率靈敏度應該根據不同的輸入訊號統計值組合來測量才會有較好的精準度。經由這樣的發現，運用在已知輸入訊號序列的情況下，我們可以利用功率靈敏度來當做功率消耗的指標，進而對輸入訊號序列進行分組的動作。因為在已分組的輸入訊號序列做取樣模擬，不但可以大幅縮短模擬全部輸入訊號序列可能需要的冗長時間，還可以避免因為取樣時的樣本失真而造成的誤差。除了已知輸入訊號序列的功率估測外，本論文還將它延伸到未知輸入訊號序列的功率估測。由於未知輸入訊號序列的功率分佈範圍廣泛，將功率分佈的情況以長條統計圖的方式顯示出來是最不失真的方式。但是因為未知輸入訊號序列的輸入組合幾乎有無限多種，所以如何有效的對輸入訊號組合分類及取樣就更形重要。

本論文所提出的自動化功率分佈圖化器可以針對無限長的輸入訊號來進行輸入訊號組合的分類及取樣。並且和線路模擬器 SPIICE 完整結合，只要給定一個電路及其輸入端的定義，本論文提出的圖化器就可以利用線路模擬器的部份模擬結果進行全自動的功率靈敏度分析來做為輸入訊號組合的分類及取樣的參考，以期能以最短的時間模擬最少的輸入訊號組合，求得該線路的功率消耗分佈並以數據及圖像方式表現。

Abstract

As the semiconductor technology getting advances, the density of the devices and the metal lines are growing too large to keep the heat conduction problem unnoticed. Furthermore, the heat generated by the circuits is boosted with the ramping of the operating speed. Improper heat conduction can lead to the smoking of the chip, torturing or breaking of the metal lines, or shifting of the performance. All these problems can be prevented by the power consumption estimation and optimization before taping out the chip.

There are many ways of estimating power corresponding to different purposes. For example, energy consumption must be evaluated for battery endurance, the power consumption is needed for temperature analysis, and the average current must be estimated for wire width design, and the current density for reliability analysis. Although these measurements are all clearly defined, but they have a common problem that they are input signal dependent.

To do the power analysis without lost of generality, an automatic power profiler integrated with the most accurate circuit simulator – SPICE is proposed. With the power profiler, users can get a visual figure of the power consumption distribution instead of numbers with uncertainties.

In this dissertation, the relation between input statistics and the power consumption of the integrated circuits is first analyzed. The power sensitivities of inputs are proven to be effective provided the nominal points are selected properly. With this knowledge, the power sensitivity sum of each input can be used to indicate the power consumption tendency of the input vectors, and to stratify the input vectors with. After the stratification, the sample variance can be reduced when simulating the input vectors selectively. In addition, stratification with power sensitivity is found to be able to prevent the pre-matured samples when estimating the average power consumption with Monte Carlo method. Besides, a new way of stratification that is suitable for stratifying infinitely long input sequences based on POST is also proposed. By putting these findings together, the SPICE circuit simulator is modified to be a tool that can visualize the distribution of the power consumption according to the user specified input statistics or input sequences.

誌 謝

我要對我的指導教授周景揚博士致上最誠摯的感謝及最高的敬意。沒有周老師的耐心指導與大力幫忙，我沒有辦法完成這個我曾經認為無法完成的事。周老師不但在研究上給我許多深入且準確的建議，他在健康、工作及家庭上的平衡更是我希望能夠達到的理想。

我也要謝謝我的另一位指導教授沈文仁博士。雖然他沒有來的及看到我的畢業論文，但是我還是希望能和他分享我畢業的榮耀與欣喜之情。沈老師對環境的敏銳觀察力以及設定目標後強勢的執行力在無形中不斷的在鞭策著我努力向前。

我還要謝謝我的老婆秀芬，在我唸博士班的這段時間，她從我的女朋友變成我的老婆，還變成了我的可愛的女兒惟惟的媽媽。因為我的論文及工作關係，在懷孕過程中無法每次陪她去產檢，生產後很多事情也都要她自力完成。即便如此，她還要當我的精神支柱，讓我在壓力過大時有人可以分？，所以我要將完成這篇論文的成就獻給她。

謝謝我的爸爸、媽媽對我的栽培，還有這段時間在經濟上及精神上給我的幫忙。雖然我的升學考試一路順遂，但是我的研究所求學過程卻讓他們比別人多擔心了好多年。也謝謝我的岳母給我及秀芬許多生活上的照顧，她總會在我想偷懶的時候激發我的鬥志。

我還要謝謝實驗室的許文俊學長教我的工作站管理、林景源學長和我討論功率估測的原理、黃俊達學長教我的進階 C 程式撰寫及 DDD 使用。還有其他實驗室的同學及學弟妹們王成業、王俊堯、江蕙如、呂炳松、涂尚璋、黃自立、許智揚、陳勇仁、曾智謀、溫衍權以及劉建男，他們在我博士論文中都給我直接的幫忙或在 meeting 時給我不少的？發及建議。

謹將這篇論文獻給所有曾經對我的博士論文有過幫助或關心的人。

Table of Contents

| | |
|--|-----------|
| 中文摘要 | i |
| Abstract | ii |
| 誌 謝 | iii |
| Table of Contents | iv |
| Figures Captions | vi |
| Table Captions | vii |
| 1 Introduction..... | 1 |
| 1.1 Electromigration | 1 |
| 1.2 Heat Conductance | 2 |
| 1.3 Wire Temperature | 4 |
| 1.4 Current Density Limit | 7 |
| 1.5 Substrate Temperature | 8 |
| 1.6 Probability Based Method [25]~[77] | 12 |
| 1.7 Simulation Based Methods[78]~[108]..... | 14 |
| 1.8 Dissertation Outline | 15 |
| 2 Study of Power Sensitivity | 16 |
| 2.1 Terminologies | 17 |
| 2.2 Relation Between p and d | 17 |
| 2.3 Power Sensitivity..... | 18 |
| 2.4 Nominal Point Selection..... | 21 |
| 2.4.1 Nominal 0 | 21 |
| 2.4.2 Analysis of Estimation Error | 24 |
| 2.5 Experimental Results | 28 |
| 2.6 Summary..... | 30 |
| 3 Bootstrap Monte Carlo with Adaptive Stratification..... | 32 |

| | |
|---|-----------|
| 3.1 Related Works | 32 |
| 3.2 Preliminary | 34 |
| 3.2.1 Normal Distribution and Gaussian Distribution | 34 |
| 3.2.2 Monte Carlo Method..... | 35 |
| 3.2.3 Bootstrap..... | 36 |
| 3.3 Bootstrap Monte Carlo Simulation..... | 38 |
| 3.3.1 Bootstrap Confidence Level..... | 38 |
| 3.3.2 Bootstrap Monte Carlo Method..... | 38 |
| 3.3.3 Bootstrap Monte Carlo vs. Conventional Monte Carlo..... | 39 |
| 3.4 Adaptive Stratified Random Sampling | 42 |
| 3.4.1 Single Variable Linear Regression | 42 |
| 3.4.2 Multiple Regression | 42 |
| 3.4.3 Variable Selection..... | 43 |
| 3.4.4 BMC with Adaptive Stratification (BMCAS) | 44 |
| 3.5 Experimental Results | 47 |
| 3.6 Summary..... | 49 |
| 4 Automatic Power Profiler..... | 51 |
| 4.1 Problem Definition | 51 |
| 4.2 Number of Samples | 53 |
| 4.3 POST - Power Sensitive Transition..... | 54 |
| 4.4 Stratification with POST..... | 58 |
| 4.5 Integration with SPICE3 | 60 |
| 4.5.1 Transient Analysis Flow | 60 |
| 4.5.2 Automatic Power Profiler..... | 62 |
| 4.6 Screen Shots..... | 63 |
| 4.7 Summary..... | 66 |
| 5 Conclusion and Future Work | 67 |
| 6 References..... | 69 |



Figures Captions

| | |
|--|----|
| Figure 1: Forces on an atom of metal..... | 1 |
| Figure 2: Flip-chip package and wire-bond packaging | 4 |
| Figure 3: Cross-section of the metal line along the wire length..... | 5 |
| Figure 4: Temperature profile of the metal line..... | 6 |
| Figure 5: Heat flows of a cubic in the substrate | 8 |
| Figure 6: Temperature distribution..... | 11 |
| Figure 7: Relationship between p_x and d_x | 18 |
| Figure 8: Simulated power vs. estimated power with N_0 | 26 |
| Figure 9: Simulated power vs. estimated power with N_{q1} | 27 |
| Figure 10: Simulated power vs. estimated power with N_{q3} | 27 |
| Figure 11: Simulated power vs. estimated power with 3-point model..... | 30 |
| Figure 12: Pseudo code of BMC | 39 |
| Figure 13: Pseudo code of BMCAS | 46 |
| Figure 14: Example of power distribution..... | 52 |
| Figure 15: Stratification with POST | 58 |
| Figure 16: Flowchart of transient analysis in SPICE..... | 61 |
| Figure 17: Flowchart of power profiler | 63 |
| Figure 18: Screen log of power profiler | 64 |
| Figure 19: Power profile from power profiler | 65 |

Table Captions

| | |
|---|----|
| Table 1: Electrical variables vs. thermal variables | 2 |
| Table 2: Properties of selected metals | 7 |
| Table 3: Levels of abstract..... | 9 |
| Table 4: The accuracy of nominal 0 | 23 |
| Table 5: Selected 3-point model VS. random 5-point model..... | 29 |
| Table 6: Conventional Monte Carlo vs. Bootstrap Monte Carlo with $\alpha=0.05$ | 39 |
| Table 7: Conventional Monte Carlo vs. Bootstrap Monte Carlo with $\alpha=0.025$ | 40 |
| Table 8: Conventional Monte Carlo vs. Bootstrap Monte Carlo with $\alpha=0.005$ | 40 |
| Table 9: Comparison of stratification method, confidence level = 90% | 48 |
| Table 10: Comparison of stratification method, confidence level = 95% | 48 |
| Table 11: Comparison of stratification method, confidence level = 99% | 49 |
| Table 12: POST and non-POST combinations..... | 54 |
| Table 13: POST identification example | 57 |
| Table 14: Sample variance differences derived from Table 13 | 57 |

1 Introduction

As the semiconductor technology getting advances, the density of the devices and the metal lines are growing too large to keep the heat conduction problem unnoticed. Furthermore, the heat generated by the circuit is boosted with the ramping of the operating speed. Improper heat conduction can lead to the smoking of the chip, torturing or breaking of the metal lines, or shifting of the performance. All these problems can be prevented by the power consumption estimation and optimization before taping out the chip. Before discussing how the power estimation can be done, the physics of electromigration will be introduced to see what measurements are demanded when doing power estimations.

1.1 Electromigration

Electromigration failure is caused by the mass transport of a metal. The force on an atom of the metal is the combination of the $F_{scatter}$ by the momentum transferred from the free electrons speeding in the metal, and the force from electric field F_{field} :

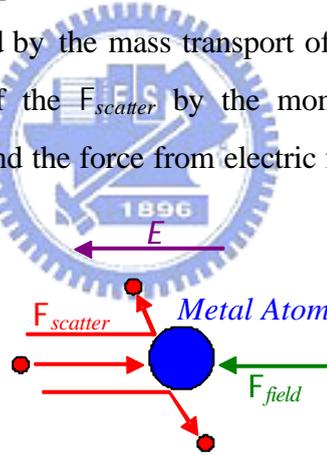


Figure 1: Forces on an atom of metal

$$F_{total} = F_{scatter} + F_{field} \cong qZ^*E . \quad (1)$$

The qZ^* is called the effective charge of the atom and is usually negative due to the dominance of the $F_{scatter}$. By the *Einstein's relation* (Appendix A), the mobility of the atom is:

$$m = \frac{qZ^*D}{k \cdot T} . \quad (2)$$

The drift velocity of the metal atom is:

$$v_d = m \cdot E = \frac{qZ^*D}{k \cdot T} \cdot r \cdot J , \quad (3)$$

where the resistivity of the metal r , the density of the current flowing in the metal J , and the diffusion coefficient D is generally given by the *Arrhenius relation*[16]:

$$D = D_0 \cdot e^{-\frac{E_a}{kT}}, \quad (4)$$

where D_0 is the diffusion coefficient and E_a is the activation energy depending on the structure and the material of the metal. It is about 0.6eV for aluminum, and 0.9eV for aluminum alloyed with 0.2% copper. The time required by the lattice atoms to migrate a certain length to cause a failure on the metal line is thus proportional to the reciprocal of v_d :

$$Time \propto \frac{k \cdot T}{qZ^* D_0 \cdot r} \cdot J^{-1} \cdot e^{\frac{E_a}{kT}}. \quad (5)$$

Equation (5) is generalized by *Jim Black* experimentally into equation (6) as the *Black's equation* defining the *Mean Time to Failure (MTF)* [13]:

$$MTF = AJ^{-n} e^{\frac{E_a}{kT}}, \quad (6)$$

where A is a constant depending on the process and the metal material, and n can be obtained by filtering the experimental results, normally around 1 to 2. With a given process, equation (6) gives a direct estimation of the endurance of a metal wire as a function of the current density flowing through and the working temperature of it. Experiments conducted by other researchers found that the *MTF* is better estimated with the average current density, J_{avg} , then RMS current density [17].

1.2 Heat Conductance

There are three ways of heat transfer - conduction, convection and radiation. Since an integrated circuit is usually sealed in a hermetic package, the heat of the chip is mainly transferred through the conduction. The variables for thermal conduction are defined similarly to the electrical conduction variables:

Table 1: Electrical variables vs. thermal variables

| Name | Electrical Conduction | | Thermal Conduction | |
|---------------------|-----------------------|--------|--------------------|--------------|
| | Symbol | (Unit) | Symbol | Symbol(Unit) |
| Voltage/Temperature | V | (Volt) | T | (°K) |

| | | | | |
|---------------|----------|--------------|----------------|---------------|
| Charge/Heat | Q | (Coulomb) | H | (Joule) |
| Current/Power | I | Q/t=(Ampere) | P | H/t=(Watts) |
| Resistance | R | V/I=(Ohms) | R _T | T/P=(°K/Watt) |
| Conductivity | s | (1/Ohms m) | k | (Watt °K m) |

The electrical conductivity is defined in equation (7). We rewrite it with the variables in Table 1 as the coefficient between current per unit area and the gradient of voltage:

$$\mathbf{s} = \frac{-I/A}{dV/dx} \quad (7)$$

The thermal conductivity is defined as the coefficient between the heat loss rate per unit area and the gradient of temperature:

$$\mathbf{k} = \frac{-P/A}{dT/dx} \quad (8)$$

There is a relation between the thermal conductivity and electrical conductivity called *Wiedemann-Franz Law* [14], that defines the *Lorenz number* L as:

$$L = \frac{\mathbf{k}}{\mathbf{s}T} = \frac{\mathbf{p}^2 k^2}{3e^2} \quad (9)$$

where k is the *Boltzmann's constant*, and e is the charge of an electron. *Wiedemann-Franz Law* shows a constant relation between the thermal conductivity and the product of electrical conductivity and the temperature.

The working temperature of a metal interconnect depends on how much heat being generated and the heat conductivity of the ambiance that are thermally connected to it. There are two different packaging schemes that induce different heat conduction paths. They are wire-bond packaging and flip-chip packaging:

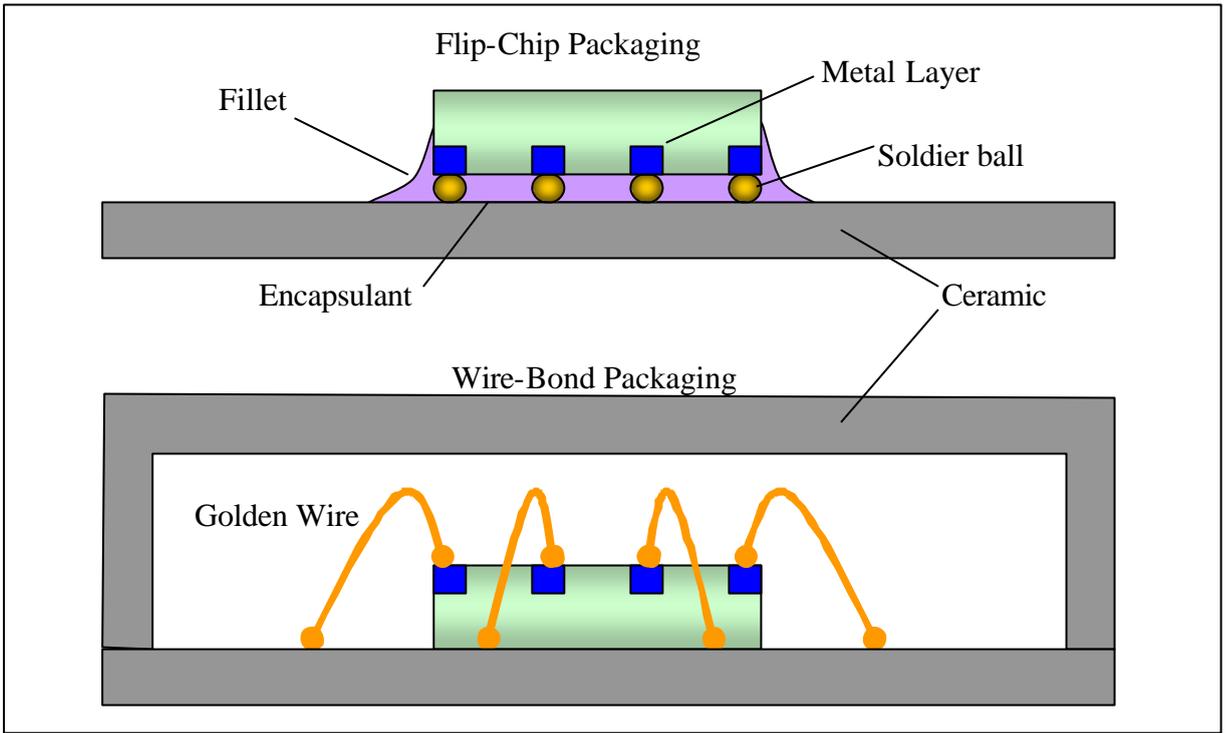


Figure 2: Flip-chip package and wire-bond packaging

For Wire-Bond packaging, the heat in the metal layers can only be conducted through the substrate which has much larger thermal conductivity than the air. Therefore, the temperature of the metal layers is usually several tens of degrees higher than the substrate. On the other hand, for the flip-chip packaging, the heat in the metal layers can be conducted through the encapsulant, and the heat of the substrate can also be conducted efficiently by external heat sink since both sides of the chip are attachable. Either way, the temperature is usually higher in the metal layer than in the substrate. The wire temperature is analyzed based on this assumption that the substrate temperature is constant in a confined local area [5][6][7][8][9], from which the analysis of wire temperature is digested and discussed in the following section.

1.3 Wire Temperature

Currents flowing through the metal will generate heat due to the collision of free carriers. The average power dissipated of the wire resistance R_{wire} is defined as:

$$P_{wire} = \frac{1}{T} \int_0^T I^2(t) R_{wire} dt = I_{rms}^2 R_{wire} \quad (10)$$

The dissipated power will heat the wire itself and put the wire under the risk of electromigration, or resistivity increase.

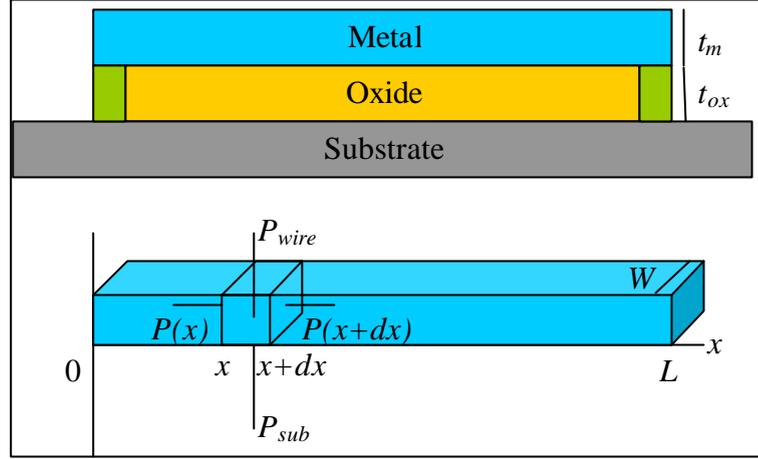


Figure 3: Cross-section of the metal line along the wire length

Figure 3 depicts a metal line with thickness t_m , width W and length L , lying on the oxide with thickness t_{ox} and connecting to the substrate with vias at both end of the line. The temperature of the wire at position x can be derived by the *first law of the thermal dynamics* based on energy conservation:

$$P(x) + P_{wire} = P(x+dx) + P_{sub}, \quad (11)$$

where $P(x)$ and $P(x+dx)$ are the heat conduction rates along the metal at position x and $x+dx$ respectively. P_{sub} is the heat conduction rate from the wire to the substrate through the oxide. By applying equation (8) and equation (11):

$$P(x) + I_{rms}^2 \frac{\mathbf{r}(x)dx}{A} = P(x+dx) - \mathbf{k}_{ox}(T) \frac{T_{ref}(x) - T(x)}{t_{ox}} W dx, \quad (12)$$

where $T_{ref}(x)$ is the temperature of the substrate at position x . By moving $P(x)$ and $P(x+dx)$ to one side and the rest to the other side, and applying equation (8):

$$\frac{I_{rms}^2 \mathbf{r}(x)}{A} + \mathbf{k}_{ox}(T) \frac{T_{ref}(x) - T(x)}{t_{ox}} W = \frac{dP(x)}{dx} = \frac{d\left(-A\mathbf{k}_m(T) \frac{dT(x)}{dx}\right)}{dx} \quad (13)$$

To simplify the analysis, the thermal conductivities \mathbf{k}_{ox} and \mathbf{k}_m are assumed to be independent of T , because it is proportional T/r , where \mathbf{r} increases approximately linearly with the temperature growth.

$$\frac{I_{rms}^2 \mathbf{r}(x)}{A} + \mathbf{k}_{ox} \frac{T_{ref}(x) - T(x)}{t_{ox}} W = -A\mathbf{k}_m \frac{d^2T(x)}{dx^2} \quad (14)$$

Dividing both side with A and move all terms to the left hand side, we get:

$$\mathbf{k}_m \frac{d^2 T(x)}{dx^2} + J_{rms}^2 \mathbf{r}(x) + \mathbf{k}_{ox} \frac{T_{ref}(x) - T(x)}{t_m t_{ox}} = 0. \quad (15)$$

By expanding $\mathbf{r}(x)$ with the resistivity equation in Appendix A around $T_{ref}(x)$:

$$\mathbf{k}_m \frac{d^2 T(x)}{dx^2} + J_{rms}^2 \mathbf{r}_{ref} [1 + \mathbf{a}_{ref} (T(x) - T_{ref}(x))] + \mathbf{k}_{ox} \frac{T_{ref}(x) - T(x)}{t_m t_{ox}} = 0 \quad (16)$$

After some arrangements, we have:

$$\frac{d^2 T(x)}{dx^2} = I^2 T(x) - I^2 T_{ref}(x) - \mathbf{q}, \quad (17)$$

where

$$I^2 = \frac{1}{\mathbf{k}_m} \left(\frac{\mathbf{k}_{ox}}{t_m t_{ox}} - J_{rms}^2 \mathbf{r}_{ref} \mathbf{a}_{ref} \right) \quad (18)$$

and

$$\mathbf{q} = \frac{J_{rms}^2 \mathbf{r}_{ref}}{\mathbf{k}_m}. \quad (19)$$

For short local wires, the substrate temperature $T_{ref}(x)$ can be assumed to be a constant T_{ref} . With this assumption, the ordinary differential equation can be solved by *Laplace transform* with the boundary conditions that $T(0) = T(L) = T_{REF}$ to get the working temperature of the wire at position x as:

$$T(x) = T_{ref} + \frac{\mathbf{q}}{I^2} \left(1 - \frac{\sinh I(L-x) + \sinh Ix}{\sinh IL} \right). \quad (20)$$

The plot of equation (20) is depicted in Figure 4 with the bold line:

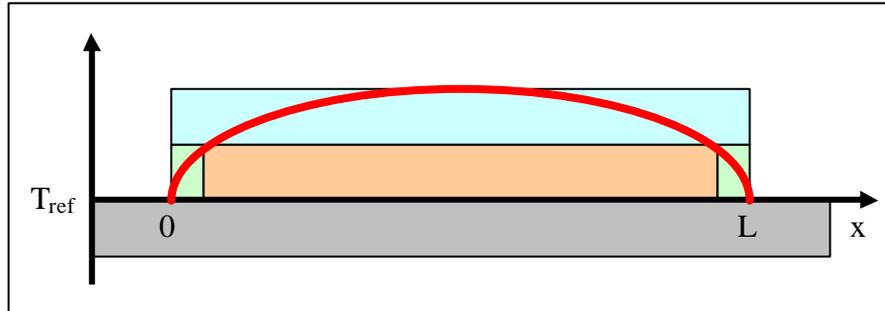


Figure 4: Temperature profile of the metal line

From Figure 4, the working temperature of the metal line can be found to have the maximum value at the position of $L/2$, and it equals:

$$T(L/2) = T_{ref} + \frac{\mathbf{q}}{I^2} = T_{ref} + \frac{J_{rms}^2 \mathbf{r}_{ref}}{\left(\frac{\mathbf{k}_{ox}}{t_m t_{ox}} - J_{rms}^2 \mathbf{r}_{ref} \mathbf{a}_{ref} \right)} = T_{ref} + \frac{1}{\left(\frac{\mathbf{k}_{ox}}{t_m t_{ox} J_{rms}^2 \mathbf{r}_{ref}} - \mathbf{a}_{ref} \right)}. \quad (21)$$

Surprisingly, the maximum working temperature of the metal is independent of the thermal conductivity k_m of the metal. It also clearly stated that the maximum wire temperature is built upon the temperature of the substrate.

1.4 Current Density Limit

The reasonable working temperature should be undoubtedly smaller than the melting points of the metals that are listed in the following table [15], in which, the resistivities r_{REF} and the associated temperature coefficients a_{REF} are measured with T_{REF} equals 20°C .

Table 2: Properties of selected metals

| Material | T_{melt} ($^\circ\text{C}$) | T_{melt} ($^\circ\text{F}$) | r_{ref} ($\Omega\text{-m}$) | a_{ref} ($^\circ\text{C}^{-1}$) |
|----------------------|---------------------------------|---------------------------------|---------------------------------|-------------------------------------|
| <i>Lead</i> (Pb) | 328 | 622 | 20.65×10^{-8} | 4.2×10^{-3} |
| <i>Zinc</i> (Zn) | 420 | 788 | 5.68×10^{-8} | 4.2×10^{-3} |
| <i>Aluminum</i> (Al) | 660 | 1221 | 2.65×10^{-8} | 4.2×10^{-3} |
| <i>Sliver</i> (Ag) | 962 | 1763 | 1.59×10^{-8} | 4.1×10^{-3} |
| <i>Gold</i> (Au) | 1064 | 1947 | 2.44×10^{-8} | 4.0×10^{-3} |
| <i>Copper</i> (Cu) | 1083 | 1981 | 1.673×10^{-8} | 4.3×10^{-3} |
| <i>Platinum</i> (Pt) | 1768 | 3215 | 10.62×10^{-8} | 3.0×10^{-3} |
| <i>Tungsten</i> (W) | 3422 | 6192 | 5.6×10^{-8} | 4.8×10^{-3} |

By limiting the maximum temperature with the melting point T_{melt} , equation (21) can be translated into:

$$T_{ref} + \frac{1}{\left(\frac{k_{ox}}{t_m t_{ox} J_{rms}^2 r_{ref}} - a_{ref} \right)} \leq T_{melt} . \quad (22)$$

After some deduction, we can get the limit of the J_{rms} from equation (22) can be expressed as:

$$J_{rms}^2 \leq \frac{k_{ox}}{t_m t_{ox} r_{ref} \left(\frac{1}{T_{melt} - T_{ref}} + a_{ref} \right)} = \frac{k_{ox} (T_{melt} - T_{ref})}{t_m t_{ox} r_{melt}} . \quad (23)$$

Consider an aluminum segment routed above the field oxide on a substrate with 20°C temperature. The field oxide thickness t_{ox} is normally about 500nm and decreases with the process advancing. The thickness of metal not only varies with the technology but also varies

with the layer number. For example, t_m ranges from 0.3um to 0.9um for metal 1 to metal 7 for TSMC 0.13um process [11]. The thermal conductivity k_{ox} and the substrate temperature T_{REF} can be generalized to the thermal conductivity from the wire segment to its closest neighbor layer, and the temperature of the neighbor layer. Since the above equations are derived on the basis of metal 1, the thermal conductivity k_{ox} is typically 1.3 (W/m²K) for the silicon dioxide. For aluminum, the temperature should be smaller than its melting point 660°C. Therefore, the maximum root-mean-square current density is:

$$J_{rms}^2 \leq \frac{1.3}{0.3 \times 10^{-6} \times 0.5 \times 10^{-6} \times 2.65 \times 10^{-8} \left(\frac{1}{660 - 20} + 4.2 \times 10^{-3} \right)} = 5.67 \times 10^{20} \text{ (A/m}^2\text{)}. \quad (24)$$

1.5 Substrate Temperature

The substrate temperature depends on the temperatures and the thermal conductivities of the materials that are thermally connected with the substrate. The substrate temperature T_{sub} can be expressed as:

$$T_{sub} = T_{room} + \text{Dissipated Power} \times \frac{\text{Chip Thickness}}{k_{package} \times \text{Chip Area}}, \quad (25)$$

where the $k_{package}$ is the equivalent thermal conductivity from the chip through the package to the outer world. For a cubic in the substrate, the heat transfer diagram is shown in Figure 5:

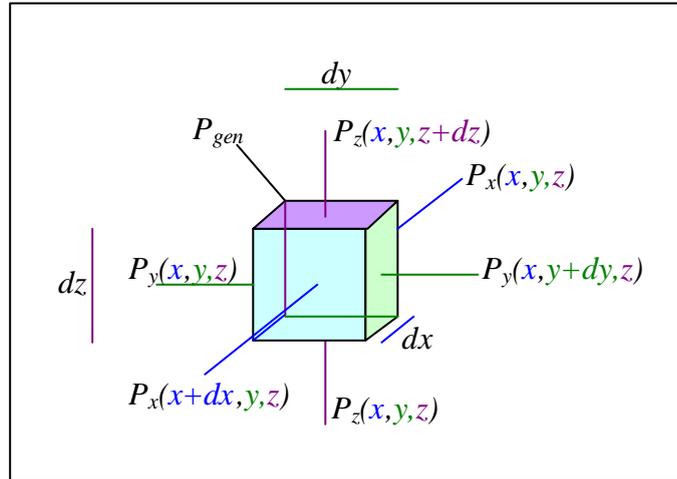


Figure 5: Heat flows of a cubic in the substrate

P_{gen} is the power dissipation of the cubic, and

$$P_x(x, y, z) = -k \frac{dT_{sub}(x, y, z)}{dx} \times dy \times dz, \quad (26)$$

$$P_y(x, y, z) = -\mathbf{k} \frac{dT_{sub}(x, y, z)}{dy} \times dx \times dz, \quad (27)$$

$$P_z(x, y, z) = -\mathbf{k} \frac{dT_{sub}(x, y, z)}{dz} \times dx \times dy. \quad (28)$$

With the *first law of thermal dynamic*, the net rate of energy flowing into the cubic plus the generation rate of the heat sources:

$$\begin{aligned} & P_x(x, y, z) - P_x(x + dx, y, z) \\ & + P_y(x, y, z) - P_y(x, y + dy, z) \\ & + P_z(x, y, z) - P_z(x, y, z + dz) + P_{gen} = 0 \end{aligned} \quad (29)$$

Dividing both side with dx , dy and dz , we get:

$$\mathbf{k} \left[\frac{\partial T^2(x, y, z)}{\partial x^2} + \frac{\partial T^2(x, y, z)}{\partial y^2} + \frac{\partial T^2(x, y, z)}{\partial z^2} \right] + PD_{gen}(x, y, z) = 0. \quad (30)$$

where PD_{gen} is the power density of the heat source. Equation (30) can be more elegant after introducing the *Laplace operator* ∇ :

$$\mathbf{k}\nabla^2 T(x, y, z) + PD_{gen}(x, y, z) = 0. \quad (31)$$

Equation (31) shows that, given the power density function PD_{gen} and the boundary conditions of the chip, the temperature function can be derived. The boundary conditions for equation (31) are modeled as the *Dirichlet condition* for the bottom surface, and the *Neumann condition* for the other five faces [3], where the *Dirichlet condition* defines a constant temperature, and the *Neumann condition* defines a constant heat flux.

However, due to the enormous number of heat sources, the numerical method to solve the exact temperature solution is too expensive to take. Not to mention that the hardness of specifying the boundary conditions. Therefore, the substrate temperature needs to be solved in a different abstract of circuit level:

Table 3: Levels of abstract

| Level of Abstract | Number of Heat Sources | $PD_{gen} \times t_{bulk}$ |
|-------------------|------------------------|--------------------------------------|
| Chip Level | 1 | Chip Power / Chip Area |
| Sub-Circuit Level | Number of sub-circuits | Sub-circuit Power / Sub-circuit Area |
| Gate Level | Number of gate | Gate Power / Gate Area |

| <i>Device Level</i> | Number of devices | <i>Device Power / Device Area</i> |
|-------------------------|------------------------|---------------------------------------|
| <i>Sub-device Level</i> | Number of heat sources | <i>Heat source / Heat source Area</i> |

Equation (25) is the simplest way to solve the substrate temperature of chip level that treats the chip as with single uniform heat source and is based on the assumption of zero heat flux for the *Neumann conditions*. However, equation (25) is too simple to model the temperature gradient due to different transition activity of the sub-circuits. Gate level thermal analysis is performed in [4], and has an error within several degrees comparing to the device level. In addition to the zero heat flux *Neumann conditions*, equation (31) is solved under the assumption that the bulk is infinitely large comparing with the area of a gate or a sub-circuit. For a sub-circuit with dimensions of $a \times b \times c$ centered at the origin, the steady state temperature rise ΔT at the observation point (x, y, z) can be defined as:

$$\Delta T(x, y, z, t) = \frac{PD_{gen}}{d \times c_p} \int_{t=0}^t G(x, a, t) G(y, b, t) G(z, c, t) dt . \quad (32)$$

where

$$G(m, n, t) = \frac{1}{2} \left[\operatorname{erf} \left(\frac{n/2 + m}{2\sqrt{g(t-t)}} \right) + \operatorname{erf} \left(\frac{n/2 - m}{2\sqrt{g(t-t)}} \right) \right], \quad (33)$$

The γ is the *thermal diffusivity* defined as

$$g = \frac{k}{d \times c_p}, \quad (34)$$

where d is the density of the material, and c_p is the specific heat of the material.

The temperature profile for the $z=0$ plan induced by a 6×10 sub-circuit heat source is depicted in the following figure:

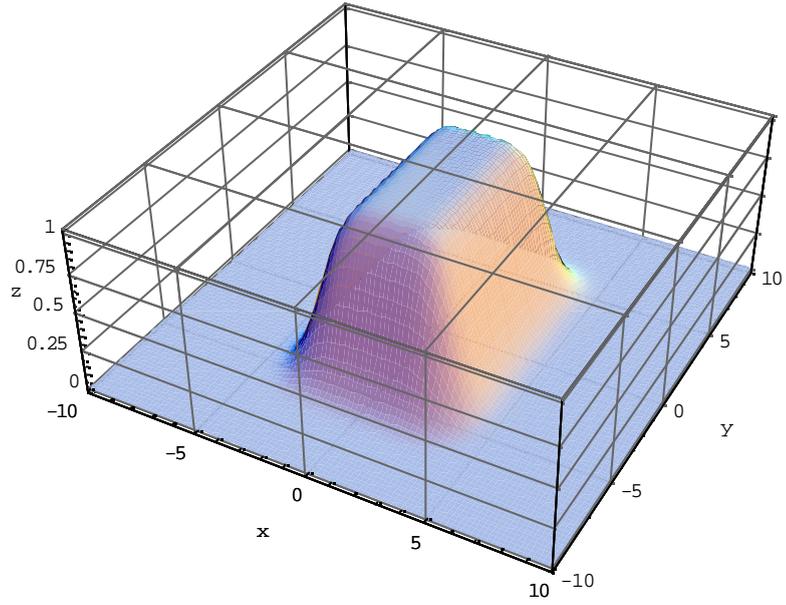


Figure 6: Temperature distribution

For a chip with n heat sources, the temperature rise of an observation point at position (x,y,z) can be obtained by summing up the temperature rises due to the n heat sources:

$$\Delta T(x, y, z, t) = \sum_{i=1}^n \Delta T_i(x, y, z, t). \quad (35)$$

From equation (35) and equation (32), to derive the substrate temperature, the only unknown parameter is the PD_{gen} of each heat sources. Therefore, this chapter can be concluded with the following remarks.

Remark 1:

The mean time to failure MTF of a wire is a function of the average current density J_{avg} and the wire temperature T_{wire} .

Remark 2:

The wire temperate T_{wire} is a function of the mean-square current density J_{rms} and the substrate temperature T_{sub} .

Remark 3:

The substrate temperature T_{sub} is a function of the heat generation rate (or the power dissipation density) PD_{gen} and the room temperature.

To sum up, reliability analysis of an integrated circuit needs three elements: Average wire current density J_{avg} , root-mean-square wire current density J_{rms} , and average sub-circuit power dissipation density PD_{gen} . A common property for these three parameters is that they are all the average values of something. Therefore, in this dissertation, the focus will be placed on the estimation and derivation of the average power consumption. The methods for measuring the power dissipation densities and current densities with simulators are detailed in the appendix.

In the following sections, the power estimation techniques proposed by other researchers will be re-examined. Power estimation techniques can be categorized as the probability based, and the simulation based, according to the utilization of simulator or not. These two categories are briefed in separated sections for the easiness of references.

1.6 Probability Based Method [25]~[77]

Probability based techniques can be categorized into some different levels. Some of them that utilize the entropy [25]~[30] are good for high-level power estimation and modeling because they are operated in the information level which is independent of the wire loading, transistor sizes, gate types or even structure independent. Going down to the gate level, most of them are estimating the signal transition probabilities because it is well known that CMOS circuit power consumption are dominated by the charging and discharging of load capacitances and the short circuit currents[1][12][10]. Early state gate level power estimation assumes that the primary inputs are spatially and temporally independent and the gates are with zero delay [31][32][33][48][54][58] and is focused on the combinational circuits. Some works extend the coverage to sequential circuits [34][35][36][65]. These techniques utilize the BDD (Binary Decision Diagram) for the switching activity estimation. The complexity grows exponentially with the increasing of number of primary inputs.

To reduce the complexity of BDD operation for switching activity estimation, the FDD (Free Decision Diagram) is proposed in [37] that constructs the BDD with additional AND and XOR nodes to reduce the complexity to polynomial time. In [38], the Boolean expressions of the switching activities are approximated by dropping some higher order terms. In [66] and [67], the authors proposed topological ways of building the BDD locally so that the number of levels in the BDD can be limited as well.

The spatial correlations are included for switching activity estimation in [39][50][55][59]. However, the complexity cost for including spatial correlation is too high to be adopted, so researchers turn to the temporal correlation for increasing the accuracy. The idea of transition density is proposed in [44]. It models the lag-1 temporal correlation with the density of an input that is making a 0 to 1 or 1 to 0 transition. It is suitable for modeling most combination circuits that are memoryless. Basing on the input probability and input transition density, a series of researches are conducted to study the sensitivity of the power consumption to the input probabilities and input transition densities [71]~[77]. Power sensitivities are defined as the sensitivity of the power consumption to the changes of the input probabilities and transition densities in [72][73][75]. It can be measured by selecting a nominal combination of the input probabilities and transition densities, and then altering one of the input probabilities or transition densities to see how power consumption will change accordingly. Power sensitivities are used not only for power estimation [74], but also for estimating maximum power [76] and building power model [77]. Based on the effectiveness of power sensitivity can be improved by carefully selecting the nominal points and proposed a new method of selecting the nominal point for measuring the power sensitivities is proposed.

In addition to the power sensitivities, more researches are developed after the definition of transition density. Many works have used it for the derivation of the switching activities [45][46]. Some others generalized the definition by giving the probability of 0 to 1 and 1 to 0 transitions different variable names called the transition probability and evaluate the transition probabilities through probability waveform simulation [40][42][52][57][60][69][70]. Although separating the transition densities into transition probabilities can model the signal characteristics more accurately, the complexity grows even higher, and researchers have been working harder finding a way out. However, using real gate delays, dealing with complex gates and the availability of input correlations are all other tough challenges for probability base approaches to solve. But these are no problems with the simulation techniques because the ability of solving the temporal and spatial correlations between the nodes in a circuit is what the simulators born with.

1.7 Simulation Based Methods[78]~[108]

Simulation based techniques utilize simulators for power estimation. Since the estimated values are obtained by simulating the circuit, the shortage of probability based techniques are overcome naturally provided the models used for simulation are accurate. The objective of simulation based techniques is to reduce the simulation time required to get the estimation done.

A direct way of reducing the time needed for simulation is to raise the level of simulation since the complexity of higher level abstract is lower. In [91], Gate level simulation is modified to incorporate the loading information so that the amount of switched capacitance can be simulated to mimic the power consumption. The authors of [94] further simplify the method by observing some primitive nodes instead of watching all of them. The approach that estimates the power between RTL level and gate level is proposed in [93]. These approaches that are built with the high level simulation are limited to the cases that high level netlist and the power simulation models are available.

Another approach is to regenerate a new input sequence that has the similar average power as that of the original input sequence. Some characteristics of the original sequence are chosen to be preserved while regenerating the shorter one including preserving the pattern transition probabilities [78][79], preserving the significant correlations between the clustered inputs [80], preserving the toggling behavior of the internal nodes [85], and preserving the spatial-temporal correlations for all inputs [104][105][106][108]. Regenerating a compact input sequence is good when the CPU time for estimating power is limited because it can regenerate the new input sequence with any length. Nevertheless, there is no error control scheme available for the regenerating process. In other words, there is no guarantee for the accuracy of the regenerated sequence.

The *Monte Carlo* approach for power estimation is proposed in [96]~[100]. The method estimates the average power by sampling the input vectors with certain length l from the original sequence and fed them into the simulator to derive a sample value of the average power. The average power consumption can be estimated with the average of several sample values. From *Central Limit Theorem (CLT)*, the sample values can be presumed as a *Normal* distribution when l approaches infinity. The probability that the estimated mean is within a certain error range of the real mean can also be derived under the assumption. The problem of

how to derive the required l for preserving the normality of \mathbf{x} is left open, and the *Bootstrap Monte Carlo* is introduced in this dissertation for solving this issue.

Sampling techniques can be further optimized through stratification of the population. Proper stratification of the population can reduce the sample variance and thus the number of sampled input vectors. To stratify the input vectors, various indicator functions are used for indicating the possible value of the power consumption corresponding to each input vector. In [92], the transition of primary inputs, primary outputs, latches and selective internal nodes are used as the indicator function. In [102][103][107], the zero delay gate level simulation switching condition is shown to be highly correlated with the transistor level simulated power, and thus is a good indicator.

1.8 Dissertation Outline

In this dissertation, the relation between input statistics and the power consumption of the integrated circuits is analyzed. The power sensitivities of inputs are proven to be effective provided the nominal points are selected properly in chapter 2 .

In chapter 3 , the power sensitivities sum of each input is used to indicate the power consumption tendency of the input vectors, and to stratify the input vectors with. After the stratification, the sample variance can be reduced when simulating the input vectors selectively. In addition, stratification with power sensitivity is found to be able to prevent the pre-matured estimation when estimating the average power consumption with *Monte Carlo* method. By introducing the Bootstrap resampling method to prevent the possible normality defect of *Monte Carlo* method, a sampling technique designated as *Bootstrap Monte Carlo* method with adaptive stratification is also proposed to provide a fast and accurate way for power estimation.

In chapter 4, a new way of stratification that is suitable for stratifying infinite length input sequences is proposed based on POST. Putting these findings together, the SPICE circuit simulator is modified to be a tool that can visualize the distribution of the power consumption according to the user specified input statistics or input sequences.

In chapter 5, this dissertation will be concluded by discussing the applications and limitations, and some possible extensions of this dissertation will be pointed out.

2 Study of Power Sensitivity

As discussed in the previous chapter, the substrate power is dissipated in the resistive devices when there are currents flowing. In digital CMOS circuits, the major current flows occur when the logic values are changing. When the output of a logic gate is changing from 0 to 1, the loading capacitor at the output will be charged and the power is dissipated in the charging network and some energy will be stored in the capacitor. The stored energy will be dissipated when the logic value is changing from 1 to 0 and the capacitor gets discharged. Analyzing the switching activities of the nodes in a circuit is appropriate for calculating steady state power consumption instead of simulating the circuit for an infinite length of time. However, the complexity of calculating the transition probability for each node grows exponentially with the size of a circuit.

Regression method had been used to construct a high level power model as a function of weighted input and output transitions in [83] and [101]. It is a good and fast estimation for randomly generated input signals, i.e., both the signal probabilities and the transition densities are around 0.5. However, cases with highly biased input probabilities could make the estimation of power far from accuracy.

In [63], the power consumption of a circuit is modeled as a function of the mean values of input signal probabilities, input transition densities and output transition densities. A power lookup table is built and indexed with those three values. Good accuracy can be obtained when the three values of the circuit under estimation are close to the pre-selected indices and the variances of the signal probability and the transition density are small. However, the output transition densities can not be calculated easily from the input statistics. Therefore, to explore the output density space requires simulating a tremendous amount of combinations of the input probabilities and the transition densities. It is very time consuming.

The concept of power sensitivity first proposed in [72], in which power is modeled as the sum of weighted uncertainties of the input signal probabilities and the transition densities plus the power at a nominal point. The weights of those uncertainties are called the power sensitivities. A Statistical Technique to Estimate Power Sensitivity abbreviated to STEPS [72] was proposed to determine power sensitivities. Another Symbolic Technique to Obtain Power

Sensitivities named STOPS [77] expresses the signal probability and the transition density of each node symbolically to determine the power sensitivities. The estimated power based on this kind of methods strongly depends on the selected nominal points. However, it is difficult to select the nominal points. The authors in [72][77] randomly selected some points but did not evaluate the accuracy of their models.

In this chapter, the correlation between power consumption and power sensitivities will be examined, and a new method of nominal point selection for measuring power sensitivities will be proposed. In addition, the importance of a good nominal point to the accuracy of the power sensitivities will also be demonstrated.

2.1 Terminologies

Signal Probability p

A digital signal at node x , $x(t)$, is either 1 or 0 if the rise/fall time and over/under shoots are neglected [13]. The expected value of a signal to be 1 in a clock cycle with a period T can be defined as

$$p_x = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt \quad (36)$$

where p_x is the equilibrium probability of $x(t)$ or the signal probability of node x .

Transition Density d

Four types of transition that a signal can make between two consecutive periods are 00, 01, 10 and 11. The transition density of $x(t)$ is defined as

$$d_x = \lim_{T \rightarrow \infty} \frac{n_x(T)}{T} \quad (37)$$

where $n_x(T)$ is the number of 01 and 10 transitions of $x(t)$ in the time interval $(-T/2, +T/2)$ [13].

2.2 Relation Between p and d

Intuitively, the 01 and 10 transition probabilities are defined as

$$\begin{aligned} p_x^{01} &= \lim_{T \rightarrow \infty} \frac{n_{x01}(T)}{T}, \\ p_x^{10} &= \lim_{T \rightarrow \infty} \frac{n_{x10}(T)}{T}, \end{aligned} \quad (38)$$

where $n_{x01}(T)$ and $n_{x10}(T)$ are the number of 01 and 10 transitions of $x(t)$ in the time interval $(-T/2, +T/2)$, respectively. Note that a digital circuit always makes a 10/01 transition at some time after a 01/10 transition. When T approaches infinity, the transition probabilities of 01 and 10 will converge to the same value. With the above definitions, four transition probabilities are derived and expressed as,

$$\begin{aligned} p_x^{00} &= 1 - p_x - d_x / 2, \\ p_x^{01} &= p_x^{10} = d_x / 2, \\ p_x^{11} &= p_x - d_x / 2. \end{aligned} \quad (39)$$

where $0 \leq p_x^{ij} \leq 1, \forall i, j \in \{0, 1\}$. A relationship between p_x and d_x is derived [63] as,

$$\frac{d_x}{2} \leq p_x \leq 1 - \frac{d_x}{2}. \quad (40)$$

The transition probabilities are a general form of p_x and d_x , therefore transition probabilities look better in equations. But equilibrium probability and transition density are more suitable for analyzing than transition probabilities, since the transition probabilities are not mutually independent. The valid combinations of (p_x, d_x) form a triangle as shown in Figure 7. The three boundary lines $d_x \geq 0$, $p_x - d_x/2 \geq 0$ and $1 - p_x - d_x/2 \geq 0$ are induced by the fact that a probability should be no less than zero. The three corners of the triangle are $(0,0)$, $(1,0)$ and $(0.5,1)$.

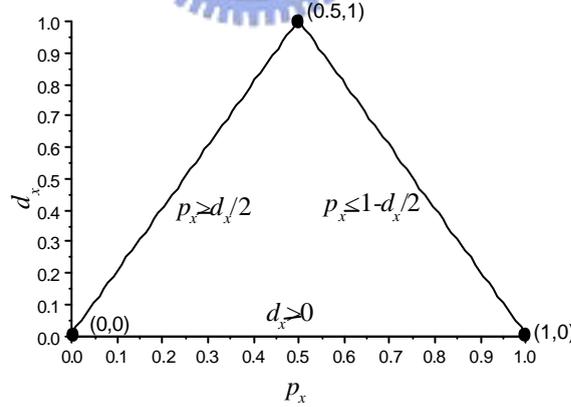


Figure 7: Relationship between p_x and d_x

2.3 Power Sensitivity

For an n -input CMOS circuit, the logic value of the i^{th} input node at time T is denoted as b_i^T . The input pattern at time T can be expressed as the transpose of $[b_0^T \ b_1^T \ \dots \ b_{n-1}^T]$. The input pattern transition of two consecutive clock periods is denoted as $V^0 V^T$,

$$V^0V^T = \begin{bmatrix} b_1^0 & b_1^T \\ b_2^0 & b_2^T \\ \vdots & \vdots \\ b_n^0 & b_n^T \end{bmatrix} \quad (41)$$

The average power consumption $Power_{avg}$ of a CMOS circuit can be expressed as

$$Power_{avg} = \sum_{\forall V^0V^T} \Pr(V^0V^T) \times Power(V^0V^T) \quad (42)$$

where $\Pr(V^0V^T)$ is the probability of the input pattern transits from V^0 to V^T , and $Power(V^0V^T)$ is the corresponding power consumption. To build a power consumption model with equation (42) needs to build a table of all the $Power(V^0V^T)$ which is absolutely infeasible for an IP block with a number of inputs larger than 32 at present.

Assume that the transitions of different inputs are independent, the $\Pr(V^0V^T)$ will be equal to the product of the independent input transition probabilities.

$$Power_{avg} = \sum_{\forall V^0V^T} \prod_{i=1}^n p_i^{b_i^0 b_i^T} \times Power(V^0V^T) \quad (43)$$

where $p_i^{b_i^0 b_i^T}$ is the probability that the logic value of the i^{th} node being b_i^0 in the beginning and b_i^T at time T . Let S be a matrix of input transition probabilities or a matrix of input probabilities and transition densities,

$$S \equiv \begin{bmatrix} p_1^{00} & p_1^{01} & p_1^{10} & p_1^{11} \\ p_2^{00} & p_2^{01} & p_2^{10} & p_2^{11} \\ \vdots & \vdots & \vdots & \vdots \\ p_n^{00} & p_n^{01} & p_n^{10} & p_n^{11} \end{bmatrix} \equiv \begin{bmatrix} p_1 & d_1 \\ p_2 & d_2 \\ \vdots & \vdots \\ p_n & d_n \end{bmatrix} \quad (44)$$

Replace the transition probabilities in Equation (43) with input statistic matrix S , and apply *Taylor's* expansion to equation (43) around a nominal input statistic matrix S_{nom} , the average power consumption of S equals,

$$\begin{aligned}
& Power_{avg}(S) \\
&= Power_{avg}(S_{nom}) + \sum_{\forall p_i^{b_i^0 b_i^T}} \frac{\partial Power_{avg}}{\partial p_i^{b_i^0 b_i^T}}(S_{nom}) \times \Delta p_i^{b_i^0 b_i^T} \\
&+ \sum_{\forall p_i^{b_i^0 b_i^T}, p_j^{b_j^0 b_j^T}} \frac{\partial^2 Power_{avg}}{2! \partial p_i^{b_i^0 b_i^T} \partial p_j^{b_j^0 b_j^T}}(S_{nom}) \times \Delta p_i^{b_i^0 b_i^T} \times \Delta p_j^{b_j^0 b_j^T} \\
&+ \dots
\end{aligned} \tag{45}$$

where $\Delta p_i^{b_i^0 b_i^T} = p_i^{b_i^0 b_i^T} - p_{i_nom}^{b_i^0 b_i^T}$, and S_{nom} is a set of nominal input statistics.

$$S_{nom} \equiv \begin{bmatrix} p_{1_nom}^{00} & p_{1_nom}^{01} & p_{1_nom}^{10} & p_{1_nom}^{11} \\ p_{2_nom}^{00} & p_{2_nom}^{01} & p_{2_nom}^{10} & p_{2_nom}^{11} \\ \vdots & \vdots & \vdots & \vdots \\ p_{n_nom}^{00} & p_{n_nom}^{01} & p_{n_nom}^{10} & p_{n_nom}^{11} \end{bmatrix} \equiv \begin{bmatrix} p_{1_nom} & d_{1_nom} \\ p_{2_nom} & d_{2_nom} \\ \vdots & \vdots \\ p_{n_nom} & d_{n_nom} \end{bmatrix} \tag{46}$$

Consider the first order approximation of equation (45), which can be expressed as

$$\hat{Power}_{avg}(S) \approx Power_{avg}(S_{nom}) + \sum_{\forall p_i^{b_i^0 b_i^T}} \frac{\partial Power_{avg}}{\partial p_i^{b_i^0 b_i^T}}(S_{nom}) \times \Delta p_i^{b_i^0 b_i^T} \tag{47}$$

The partial derivative is the changing rate of the power consumption due to the change of one input transition probability, it is called *power sensitivity* in [72][77].

The values of the partial derivatives can be estimated with a statistical method STEPS [72]. STEPS is a Monte Carlo based approach that simulates a circuit with randomly generated patterns to get samples of input power sensitivities until the mean value of those samples converges. Another method STOPS [77] for estimating power sensitivities requires topological partitioning to reduce its enormous complexity.

Since the efficiency of measuring power sensitivities is not a point here, a straightforward method is adopted for measuring them. The circuit is first simulated with the nominal input statistics S_{nom} . A small variation is then assigned to one of the transition probabilities of an input, and the simulation is proceeded again to compute the changing rate of power corresponding to the variation. The steps are repeated for other input nodes until all partial derivatives are derived.

2.4 Nominal Point Selection

2.4.1 Nominal 0

To construct a power model based on power sensitivities, one must choose some nominal points first and calculate the power sensitivities according to the chosen nominal points. Without doubt, the more nominal points are chosen the more accurate the model will become. However, adding a nominal point requires $2 \times n + 1$ more simulations to be carried out in a circuit with n inputs [72]. Furthermore, choosing nominal points arbitrarily makes the error range of the constructed power model unpredictable. It is thus important to choose the right nominal points to minimize the error of the power model with as few nominal points as possible.

This section focuses on finding a nominal input statistic matrix that estimates the average power of all kinds of S well. The average power is estimated by using equation (47), and the estimation error is defined as

$$Error = Power_{avg}(S) - \left[Power_{avg}(S_{nom}) + \sum_{\forall p_i^{b_i^0 b_i^T}} \frac{\partial Power_{avg}}{\partial p_i^{b_i^0 b_i^T}}(S_{nom}) \times \Delta p_i^{b_i^0 b_i^T} \right] \quad (48)$$

Equation (48) can be approximated with the second order term, while neglecting the higher order terms. The average error will be

$$E[|Error|] \approx E \left[\sum_{\forall i \neq j} \left(\frac{\partial^2 Power_{avg}}{2! \partial p_i^{b_i^0 b_i^T} \partial p_j^{b_j^0 b_j^T}}(S_{nom}) \times \Delta p_i^{b_i^0 b_i^T} \times \Delta p_j^{b_j^0 b_j^T} \right) \right] \quad (49)$$

The second order partial derivatives in equation (49) can be expressed as

$$\frac{\partial^2 Power_{avg}}{\partial p_i^{b_i^0 b_i^T} \partial p_j^{b_j^0 b_j^T}}(S_{nom}) = \sum_{\forall V^0 V^T \ni (b_i^0 b_i^T, b_j^0 b_j^T)} \left[\left(\prod_{\forall b_k^0 b_k^T \in V^0 V^T, k \neq i, j} p_{k_nom}^{b_k^0 b_k^T} \right) \times Power(V^0 V^T) \right] \quad (50)$$

Let p_{max} be the maximum value of $p_{k_nom}^{b_k^0 b_k^T}$, $Power_{max}$ be the maximum value of $Power(V^0 V^T)$.

The right hand term (R.H.T.) in equation (50) can be expressed as

$$R.H.T. \leq 4^{n-2} \times p_{max}^{n-2} \times Power_{max} \quad (51)$$

After substituting equation (51) into equation (49), the average error is bounded by

$$\begin{aligned}
& E[|Error|] \\
& \leq \frac{1}{2!} \times 4^{n-2} \times p_{\max}^{n-2} \times Power_{\max} \times E \left[\sum_{\forall (i,j), i \neq j} \left(\Delta p_i^{b_i^0 b_i^T} \times \Delta p_j^{b_j^0 b_j^T} \right) \right] \\
& = \frac{1}{2!} \times 4^{n-2} \times p_{\max}^{n-2} \times Power_{\max} \times E \left[\sum_{\forall (i,j), i \neq j} \left(\left| p_i^{b_i^0 b_i^T} - p_{i_nom}^{b_i^0 b_i^T} \right| \times \left| p_j^{b_j^0 b_j^T} - p_{j_nom}^{b_j^0 b_j^T} \right| \right) \right].
\end{aligned} \tag{52}$$

In equation (52), the indices of summation and expectation are independent. Therefore, the summation operator can be moved to the outside of the expectation operator,

$$\begin{aligned}
& E[|Error|] \\
& \leq \frac{1}{2!} \times 4^{n-2} \times p_{\max}^{n-2} \times Power_{\max} \times \sum_{\forall (i,j), i \neq j} \left(E \left[\left| p_i^{b_i^0 b_i^T} - p_{i_nom}^{b_i^0 b_i^T} \right| \times \left| p_j^{b_j^0 b_j^T} - p_{j_nom}^{b_j^0 b_j^T} \right| \right] \right)
\end{aligned} \tag{53}$$

Since the transition probabilities of input i and j are assumed to be mutually independent, the expected value of the product of the transition probability differences in equation (53) is equal to the product of the expectations,

$$\begin{aligned}
& E[|Error|] \\
& \leq \frac{1}{2!} \times 4^{n-2} \times p_{\max}^{n-2} \times Power_{\max} \times \sum_{\forall (i,j), i \neq j} \left(E \left[\left| p_i^{b_i^0 b_i^T} - p_{i_nom}^{b_i^0 b_i^T} \right| \right] \times E \left[\left| p_j^{b_j^0 b_j^T} - p_{j_nom}^{b_j^0 b_j^T} \right| \right] \right)
\end{aligned} \tag{54}$$

From equation (54), the expected estimation error can be minimized if the nominal input transition probabilities are suitably selected to minimize the two expected values of the differences between real input transition probability and nominal input transition probability. Generally, for an unknown distribution of input transition probability, it is reasonable to assume the distribution to be a normal or a uniform distribution, both of which are symmetric to the mean. For a distribution of input transition probability that is symmetric to the mean, the $p_{i_nom}^{b_i^0 b_i^T}$ can be chosen to be equal to the mean value of the distribution of input transition probability $E[p_i^{b_i^0 b_i^T}]$ for all i , to achieve the minimum expected error.

The following experiments are conducted to verify the above analysis. The input signals of all input nodes of a circuit are assumed mutually independent, and the transition probabilities are assumed to be uniformly distributed as a random input usually is. Both of them have the same mean value of 0.5. A nominal input statistics S_{nom} is thus built with each element equals 0.25. This nominal point will be designated as nominal 0 (N_0) here. That means, d_i is uniformly distributed in $[0,1]$, and consequently, p_i is uniformly distributed in $[d_i/2, 1-d_i/2]$.

Please note that the proposed power model can deal with any distribution since we did not make any assumption about the input statistics distribution is made in the analysis.

Five other nominal points are randomly constructed for comparison. Five hundred S 's are randomly generated following the distributions, $d_i \sim U[0,1]$ and $p_i \sim U[d_i/2, 1-d_i/2]$, where $U[a,b]$ is a uniform distribution between a and b . Two thousands input vectors corresponding to each S are randomly generated and fed into *PowerMill* to be simulated for the exact power consumption. The power consumption corresponding to each S is also evaluated by the first order approximation with nominal point N_0 and five other randomly selected nominal points. The error of the power estimated with each nominal point is defined as:

$$Error = \frac{Estimated_Power - Simulated_Power}{Simulated_Power} \times 100\%$$

$$Avg_Err = E[|Error|]$$

$$Max_Error = Max[|Error|]$$
(55)

The estimation errors are shown in Table 4:

Table 4: The accuracy of nominal 0

| Circuit | N0 | Rand0 | Rand1 | Rand2 | Rand3 | Rand4 |
|---------|-------|--------|--------|--------|--------|--------|
| Cm138a | 5.78% | 12.63% | 15.38% | 11.07% | 6.62% | 8.08% |
| Cm150a | 2.29% | 4.47% | 4.82% | 4.07% | 8.03% | 3.12% |
| Cm151a | 3.66% | 7.83% | 11.65% | 5.38% | 6.87% | 10.57% |
| Cm152a | 3.68% | 41.35% | 5.88% | 8.46% | 4.68% | 6.82% |
| Cm162a | 4.31% | 28.5% | 5.89% | 5.74% | 7.36% | 6.50% |
| Cm163a | 3.73% | 29.92% | 7.65% | 7.22% | 6.58% | 10.63% |
| Cm42a | 4.62% | 22.08% | 12.55% | 5.97% | 7.82% | 5.50% |
| Cm82a | 5.69% | 24.39% | 12.68% | 9.70% | 12.83% | 12.76% |
| Cm85a | 2.53% | 16.72% | 4.70% | 4.11% | 5.39% | 3.87% |
| Cmb | 1.82% | 3.99% | 3.22% | 3.06% | 2.75% | 3.05% |
| Comp | 2.90% | 12.05% | 4.57% | 4.53% | 5.15% | 5.60% |
| Cu | 2.62% | 4.01% | 5.74% | 5.48% | 4.65% | 5.47% |
| Decod | 4.89% | 15.84% | 14.07% | 8.49% | 9.69% | 11.06% |
| F51m | 2.45% | 10.42% | 4.56% | 3.45% | 4.32% | 3.21% |

| | | | | | | |
|---------|-------|--------|-------|-------|-------|-------|
| Average | 3.64% | 16.73% | 8.10% | 6.20% | 6.62% | 6.87% |
|---------|-------|--------|-------|-------|-------|-------|

There are seven columns in Table 4, the first column are the names of the circuits of MCNC benchmark circuits. The second column contains the results of the N_0 . The other columns are the results of five other randomly selected nominal points for comparison. The elements in this table are the average percentage of error of the estimation result with corresponding nominal point for the 500 random samples. The nominal point constructed with the mean values of input transition probabilities always gives the best estimation for all circuits under test. From the above analyses, selecting the mean values of the input transition probabilities as the elements of the nominal input statistics can achieve the minimum average error.

2.4.2 Analysis of Estimation Error

Although the average error of nominal 0 is minimized, the maximum error is still unacceptably large in some cases. For a power model, the worst-case estimation error is as important as the average error. In order to improve the accuracy, the estimation error of nominal zero should be analyzed to locate the cases that the largest estimation error occurs and to select some more nominal points to improve the power model.

The estimation error in equation (48) is a function of S_{nom} and S . As the first S_{nom} (N_0) is already decided, the estimation error of N_0 is a function of S . Before the analysis, there are some properties of equation (43) and equation (48) should be noted.

Property 1: Average power equation (43) is a linear equation corresponding to the four transition probabilities of each input. In other words, for each input within each term of the average power equation, the sum of the powers of the four transition probabilities is at most 1.

Proof: Every term in average power equation (43) is a product of the $\Pr(V^0 V^T)$ and the $Power(V^0 V^T)$, where the $\Pr(V^0 V^T)$ is a product of exactly one of the four transition probabilities from every input. The property is proved.

Property 2: The estimation error equation is a linear equation corresponding to every input transition probabilities.

Proof: Since the average power equation is a linear function corresponding to each input, and the estimation error equation is the high order derivatives part of the Taylor's expansion of average power equation, this property is proved.

The S that maximizes the estimation error will be denoted as S_{err} , and the elements of S_{err} will be denoted as p_{i_err} and d_{i_err} .

Theorem 1: The (p_{i_err}, d_{i_err}) is on one of the three points: (0,0), (0,1) and (0.5,1), that are the three extreme points of the p-d triangle.

Proof: Since the estimation error is a linear function corresponding to each input, it is a linear optimization problem finding the maximum value of estimation error corresponding to each input [81]. The solution to the linear optimization problem is at least one of the extreme points of the feasible domain. The feasible domain here is the valid p-d triangle of an input and the extreme points of the feasible domain are (0,0), (0,1) and (0.5,1).

From **Theorem 1**, we conclude that the number of possible solutions of S_{err} is 3^{n_in} , where n_in is the number of inputs. Although the number of possible solution of S_{err} is greatly reduced from a $2*n_in$ dimension space of real number to 3^{n_in} , the complexity of trying all the possible combination of (p_{i_err}, d_{i_err}) is still exponentially growing with the number of inputs. A heuristic approach of finding the S_{err} is required. The second order terms in the estimation error equation are the partial derivatives of the first order derivatives. Therefore, for an S that maximizes the sum of the first order terms, it is more likely to maximize the sum of the second order terms. The physical meaning of the sum of the first order terms is the difference between the estimated power and the nominal power. In other words, an input statistic S with an estimated power greatly larger or smaller than nominal power is most likely to have a larger estimation error.

Take circuit cm138a as an example to observe the relationship between the estimated power and the estimation error are depicted in Figure 8.

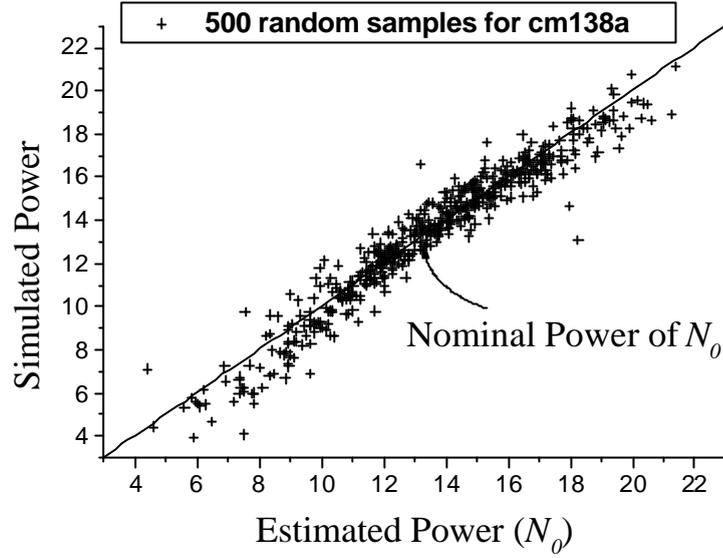


Figure 8: Simulated power vs. estimated power with N_0

The solid line in Figure 8 is the zero error line. It illustrates the ideal estimation of the simulated power. As shown in Figure 8, the estimated power deviates from the zero error line when it is away from the nominal power. That is, when the difference between the estimated power and the nominal power becomes larger, so is the estimation error. This observation supports the derived heuristic. With the heuristic, two possible S_{err} can be located. One is for the resulted largest estimated power and is denoted as S_{max} and the other, denoted as S_{min} , results in the smallest estimated power. However, S_{max} and S_{min} are not suitable for being our second and third nominal points since there are seldom S that can reach the S_{max} and S_{min} . Therefore, it is the better way to choose the second and the third nominal points as the average of N_0 and S_{max} and the average of N_0 and S_{min} . The second nominal point placed on the average of N_0 and S_{max} is designated as N_{q3} , and the third nominal point placed on the average of N_0 and S_{min} is designated as N_{q1} . The elements of the third quartile nominal point, N_{q3} are

$$\begin{aligned} p_{i_{q3}} &= (p_{i_{max}} + p_{i_{n0}}) / 2, \\ d_{i_{q3}} &= (d_{i_{max}} + d_{i_{n0}}) / 2. \end{aligned} \quad (56)$$

In the same manner, the first quartile nominal point, N_{q1} , can be obtained.

$$\begin{aligned} p_{i_{q1}} &= (p_{i_{min}} + p_{i_{n0}}) / 2, \\ d_{i_{q1}} &= (d_{i_{min}} + d_{i_{n0}}) / 2. \end{aligned} \quad (57)$$

Illustrated in Figure 9 are the experimental results of using only the nominal point N_{q1} . The estimation errors of the points in the region with relatively smaller power are obviously reduced, while others are increased.

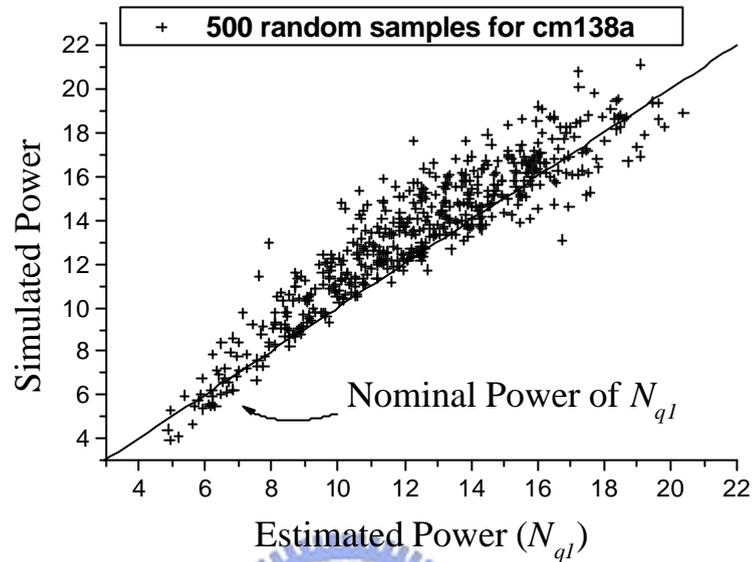


Figure 9: Simulated power vs. estimated power with N_{q1}

On the other hand, Figure 10 shows that less error is induced in the high power region when the power is estimated with only N_{q3} .

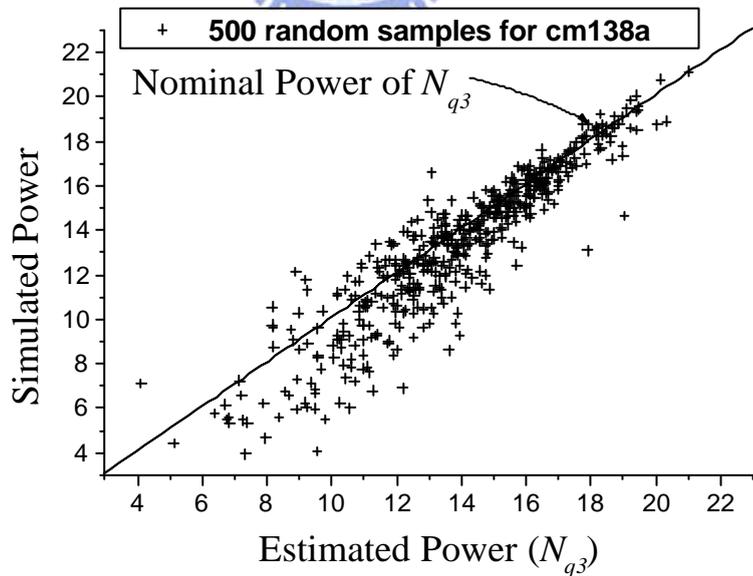


Figure 10: Simulated power vs. estimated power with N_{q3}

Since the estimation error depends on the selected nominal point, a guideline for selecting the optimal nominal point is required for minimizing the estimation error. It is possible to build a

power model constructed of these three nominal points and dynamically select the most suitable nominal point(s) to give power estimation. The details of constructing the 3-point power model will be discussed in the following section.

2.5 Experimental Results

With the three nominal points obtained in the previous section, we can construct a 3-point power model for a circuit. Considering an input statistic S , let $Power_{avg}(S)|_{N_{q1}}$, $Power_{avg}(S)|_{N_0}$ and $Power_{avg}(S)|_{N_{q3}}$ be the estimated power done with the nominal points N_0 , N_{q1} and N_{q3} , respectively. Let $Power_{avg}(N_{q1})$, $Power_{avg}(N_0)$ and $Power_{avg}(N_{q3})$ be the nominal power of N_{q1} , N_0 and N_{q3} , respectively. From previous analysis, the estimation error is likely to grow with the difference between the estimated power and the nominal power. Therefore, we take the distance between the estimated power and the nominal power as parameters of interpolation while using the proposed 3-point model. The reported estimation power from our 3-point model is evaluated with the following equations:

$$\begin{aligned} dis_1 &= Power_{avg}(S)|_{N_{q1}} - Power_{avg}(N_{q1}), \\ dis_0 &= Power_{avg}(S)|_{N_0} - Power_{avg}(N_0), \\ dis_3 &= Power_{avg}(S)|_{N_{q3}} - Power_{avg}(N_{q3}). \end{aligned} \quad (58)$$

$$Estimated_Power = \begin{cases} 1. Power_{avg}(S)|_{N_{q1}}, & \text{for } dis_1 < 0 \\ 2. \frac{Power_{avg}(S)|_{N_{q1}} \times |dis_0| + Power_{avg}(S)|_{N_0} \times |dis_1|}{|dis_0| + |dis_1|}, & \text{for } dis_0 < 0 < dis_1 \\ 3. \frac{Power_{avg}(S)|_{N_{q3}} \times |dis_0| + Power_{avg}(S)|_{N_0} \times |dis_3|}{|dis_0| + |dis_3|}, & \text{for } dis_3 < 0 < dis_0 \\ 4. Power_{avg}(S)|_{N_{q3}}, & \text{for } dis_3 > 0 \end{cases} \quad (59)$$

The following is an experiment of comparing our 3-point model with some other power model. Estimation error is calculated with equation (55). The standard deviation of estimation error is defined as:

$$STD_Error^2 = E[Error^2] - Avg_Error^2 \quad (60)$$

The model used for comparison is constructed with five randomly selected nominal points whose nominal transition probabilities are randomly generated with $d_i \sim U[0,1]$,

$p_i \sim U[d_i/2, 1-d_i/2]$. The nominal 0 model, the 3-point model and the random 5-point model are all tested with the same 500 randomly generated S_s . The average errors and maximum errors are listed in Table 5.

Table 5: Selected 3-point model VS. random 5-point model

| Circuit | Nominal 0 | | | 3-Point Model | | | Random 5-Point | | |
|---------|-----------|-------|------|---------------|-------|------|----------------|--------|-------|
| | Avg | Max | STD | Avg | Max | STD | Avg | Max | STD |
| Cm138a | 5.78 | 83.49 | 7.55 | 4.27 | 37.25 | 4.70 | 5.66 | 94.74 | 7.22 |
| Cm150a | 2.29 | 12.60 | 2.45 | 2.35 | 21.50 | 2.20 | 4.43 | 26.21 | 3.98 |
| Cm151a | 3.66 | 27.09 | 3.66 | 3.47 | 25.70 | 3.63 | 7.51 | 59.32 | 7.58 |
| Cm152a | 3.68 | 76.23 | 5.60 | 3.55 | 57.61 | 5.27 | 6.42 | 110.0% | 8.33 |
| Cm162a | 4.31 | 45.18 | 4.34 | 4.07 | 30.81 | 3.93 | 6.95 | 70.68 | 8.45 |
| Cm163a | 3.73 | 27.52 | 4.11 | 4.26 | 32.44 | 3.67 | 11.20 | 50.45 | 10.45 |
| Cm42a | 4.62 | 51.84 | 5.63 | 3.81 | 31.37 | 4.28 | 8.39 | 44.16 | 7.53 |
| Cm82a | 5.69 | 62.68 | 5.83 | 5.10 | 43.73 | 5.40 | 10.62 | 42.55 | 8.40 |
| Cm85a | 2.53 | 22.07 | 2.36 | 2.50 | 14.26 | 2.23 | 5.19 | 34.90 | 6.18 |
| Cmb | 1.82 | 8.29 | 1.57 | 1.78 | 8.83 | 1.47 | 3.01 | 13.64 | 2.55 |
| Comp | 2.90 | 14.53 | 2.39 | 3.05 | 14.92 | 2.27 | 5.35 | 33.35 | 4.90 |
| Cu | 2.62 | 17.44 | 2.67 | 2.30 | 14.48 | 2.10 | 4.36 | 39.94 | 4.68 |
| Decod | 4.89 | 40.69 | 4.84 | 3.80 | 29.80 | 3.49 | 5.52 | 39.00 | 5.49 |
| F51m | 2.45 | 20.13 | 2.32 | 1.93 | 12.38 | 1.76 | 3.60 | 16.96 | 3.41 |
| Avg | 3.64 | 36.41 | 3.95 | 3.30 | 26.79 | 3.31 | 6.30 | 48.28 | 6.37 |

In Table 5, there are 3 columns for each power model. The Avg column is the average percentage of error as in Table 4. The Max column is the maximum percentage of estimation error. The STD column is the standard deviation of the 500 estimation errors. From Table 5, we can observe that 3-point model does have smaller maximum error than nominal 0 in most cases. However, in cm150a and comp the maximum error gets worse. This is because that there are $2n$ variables for an n -inputs circuit, and therefore $2n$ dimensions in the space of the input statistics. The 500 input statistic combinations do not cover the worst corner of nominal

0. In other words, we may need more than 500 input statistics combinations to trigger the real maximum error of nominal 0 when the number of inputs is large. However, we can still observe that every value in the STD column of 3-point model is smaller than the corresponding one of nominal 0. It means that estimation errors of 3-point model are distributed in a smaller range than nominal 0. Besides, even with five nominal points, the accuracy of the random method is still far behind the proposed 3-point model.

The scatter graph of simulated power and estimated power with 3-point model is show in Figure 11.

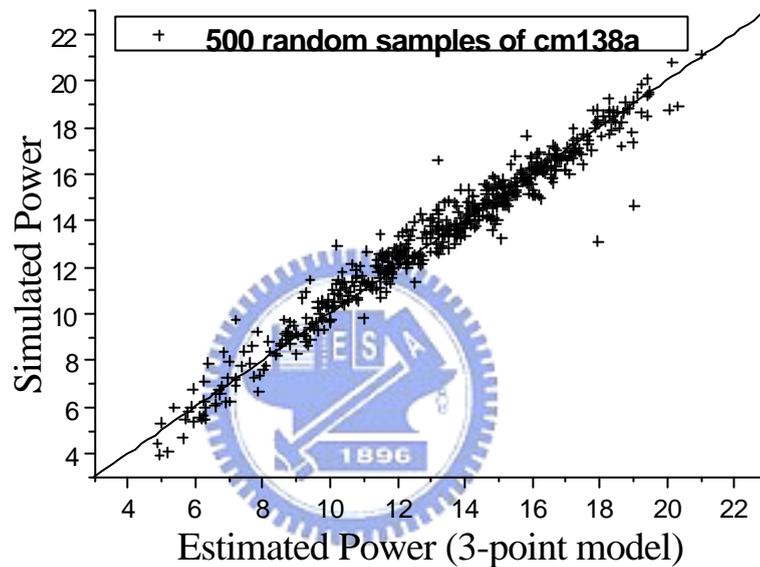


Figure 11: Simulated power vs. estimated power with 3-point model

Comparing Figure 11 with Figure 8, we can see that with our 3-point model, the small average estimation error of N_0 is kept while the maximum estimation error is minimized for cm138a.

2.6 Summary

In this chapter, a nominal point selection method for power models based on power sensitivities is proposed. By analyzing the relationship between the dynamic power consumption of CMOS circuits and their input signal statistics, a guideline of selecting the nominal point is proposed. From our analysis, the first nominal point is selected to minimize the average estimation error and two other nominal points are selected to minimize the maximum estimation error. Both the theoretical evaluations and the experimental results show

that putting the nominal point on the mean of the input transition probabilities is a very good choice. Furthermore, we propose a 3-point model to achieve even better performance. The proposed 3-point model not only keeps the small average estimation error nominal 0, but also reduces the standard deviation of the estimation error.

Since power sensitivities are derived from *Taylor's* expansion, the accuracy of the power sensitivities and the accuracy of the power estimation equation based on power sensitivities highly depend on the position of the nominal point where the *Taylor's* expansion is performed. Provided that the position of the nominal point is not too far away from the point under estimation, the power sensitivities can be a good indicator of the power consumption trend.

We will utilize the power sensitivities for indicating the power consumption of each input vector when estimating the average power consumption of a given input sequence in the following chapter.



3 Bootstrap Monte Carlo with Adaptive Stratification

In the previous chapter we proposed a method of nominal point selection for power estimation when input statistics are given. But there are more cases that users have only an input sequence of finite length, and still need a quick way of finding the power consumption or port current for reliability analysis or specification check. In such circumstances, simulation based approach is more suitable since there is no need of pre-calibrated power model. In this chapter, we will discuss how to obtain the average power without simulating the whole input sequence to speed up the estimation process.

3.1 Related Works

With the increasing size of design blocks, the number of input vectors required for estimating the power consumption of a circuit is growing exponentially. In the meantime, the time needed for simulating each input vector increases rapidly with the growing complexity of circuits. In previous literatures, methods for shortening the time required for power estimation can be classified into two categories. One is to generate a shorter input sequence, and the other is to sample a small portion of the input vectors from the original sequence. To regenerate an input sequence that has the similar average power as that of the original input sequence, some features of the original sequence need to be preserved while regenerating the shorter one. These features include preserving the pattern transition probabilities [79], preserving the spatial-temporal correlations for all inputs [108], and preserving the significant correlations between the clustered inputs [80]. Regenerating a compact input sequence sounds easy. However, the compact input sequence can only be generated according to a user-specified compaction ratio, which users usually do not know the proper value.

The Monte Carlo approach for power estimation is proposed by F. Najm [97]. The method estimates the average power by sampling the input vectors with certain length l from the original sequence and fed them into the simulator to derive a sample value of the average power. The average power consumption can be estimated with the average of several sample

values. From *Central Limit Theorem (CLT)*, the sample values can be presumed as a *Normal* distribution when l approaches infinity. The probability that the estimated mean is within a certain error range of the real mean can also be derived under the assumption. However, the required l to preserve the normality of \mathbf{x} is not discussed. If \mathbf{x} is far from a *Normal* distribution, the basis of the Monte Carlo method fails and the estimated power may have a larger error level than expected.

Bootstrap theory is a re-sampling technique that will generate Bootstrap samples by picking the sample data with replacement and report a Bootstrap confidence interval without assuming any parameter of the distribution [87]. This is also known as non-parametric Bootstrap re-sampling. By adopting the Bootstrap technique, we developed a way to calculate a more accurate confidence level to assure that the user specified confidence level would not be violated in Monte Carlo simulation.

Although the Monte Carlo method can achieve acceptable input sequence compaction ratio generally, it suffers severe degradation as dealing with power histograms like bi-modal or multi-modal [97]. For Monte Carlo approach, large sample variance means large number of samples required for the estimation to converge to the real value. The stratification method on the original input sequence is proposed to minimize the sample variance and the probability of generating the pre-matured samples [103]. According to the method, a gate level power model is required for roughly estimating the power consumption of the original input sequence on a zero-delay logic simulator. The zero-delay gate-level power consumption is used as an indicator of the circuit-level power consumption. With this indicator, the original input sequence can be partitioned into strata, within which the input vectors are with similar power consumptions such that the samples sampled from these strata can have a smaller sample variance. However, the gate level net-list sometimes needs to be concealed especially when they are the intellectual property (IP). A novel input sequence stratification technique is proposed in this dissertation. It utilizes the multiple regression method on the sampled input vectors to find the weighting of each input transition, which can be used in the power indicator function for stratification. The proposed technique can re-stratify the original input sequence according to the updated samples, and keep the sample variance the smallest.

The following parts of this chapter are organized as follows. In section 3.2, some essential definitions and bases are introduced. Section 3.3 details the concepts of the Bootstrap Monte

Carlo method and demonstrates its efficiency. In section 3.4, the proposed adaptive stratification technique with multiple regression method is presented. The flow of the Bootstrap Monte Carlo method combined with the adaptive stratification technique is shown and evaluated in section 3.4.

3.2 Preliminary

3.2.1 Normal Distribution and *Gaussian* Distribution

Definition 1: Normal random variable

A random variable (RV) x is normally distributed with mean m_x , and standard deviation s_x if its probability density function (denoted as p.d.f.) equals:

$$f(x) = \frac{1}{s_x \sqrt{2p}} e^{-(x-m_x)^2/2s_x^2}. \quad (61)$$

Definition 2: *Gaussian* (Standard Normal) random variable:

A RV y is *Gaussian* if its p.d.f. is defined as:

$$g(y) = \frac{1}{\sqrt{2p}} e^{-y^2/2}. \quad (62)$$

Definition 3: Cumulative distribution function (c.d.f.) of *Gaussian*

The probability of a *Gaussian* RV y smaller than an arbitrary value y is defined as:

$$G(y) = P\{y \leq y\} = \int_{-\infty}^y \frac{1}{\sqrt{2p}} e^{-x^2/2} dx. \quad (63)$$

Definition 4: a -percentile of *Gaussian*

The a -percentile of *Gaussian* is denoted as z_a and expressed as:

$$z_a = G^{-1}(a), \quad 0 \leq a \leq 1. \quad (64)$$

Note that the p.d.f. of the *Gaussian* RV is an even function, therefore $z_a = -z_{1-a}$, and $z_{0.5} = 0$.

Sample mean and sample variance:

Let $\{x_i, i = 1, 2, \dots, n\}$ be n randomly sampled elements out of a population with an arbitrary distribution, the *sample mean* is defined as the arithmetic average of these n samples

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (65)$$

The sample variance s^2 is defined as:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2. \quad (66)$$

3.2.2 Monte Carlo Method

The power consumption of a CMOS circuit is dominated by charging and discharging of the load capacitances at each gate output. The average power consumption can thus be defined as a function in terms of successive input patterns:

$$\mathbf{m}_x = \sum_{j=0}^{N-1} Power(V^j) / N, \quad (67)$$

where \mathbf{m}_x is the average power consumption, V^j is the input vector from the j^{th} pattern to the $(j+1)^{\text{th}}$ pattern, $Power(V^j)$ is the power measurement, and N is the number of input vectors.

Let pwr be the RV defined on a sample space containing all $Power(V^j)$. The average of l values of pwr is called a random sample x , whose sample mean approaches the desired average power, \mathbf{m}_x , and can be expressed as:

$$x = \frac{1}{l} \sum_{i=1}^l pwr_i, \quad (68)$$

where pwr_i is a value of the RV pwr . According to the Central Limit Theorem (CLT), the RV x has a distribution close to normal distribution for large l .

To estimate the \mathbf{m}_x in (67) without simulating all input vectors, the Monte Carlo approach for power estimation can help. Let \bar{x} and s^2 be the sample mean and sample variance of x , respectively. From equation (64) to (66), the following results can be derived:

$$P\left\{\left|\frac{\mathbf{m}_x - \bar{x}}{\bar{x}}\right| \leq rel_err\right\} = 1 - 2\mathbf{a}, \quad 0 \leq \mathbf{a} \leq 0.5, \quad (69)$$

where $rel_err = \frac{z_{1-\mathbf{a}} s}{\bar{x} \sqrt{n}}$.

The rel_err stands for the related error level, \mathbf{a} is the confidence level, and n is the number of samples of x . Equation (69) means that the user can have a confidence level of $1-2\mathbf{a}$ about the claim that the error between the real mean \mathbf{m}_x and the sample mean \bar{x} is smaller than the related error level. If the related error level rel_err is larger than the user specified error level \mathbf{e} , one or more samples of x should be picked and the sample mean and rel_err are evaluated again. The procedure is iterated until the user-specified error level \mathbf{e} is satisfied.

3.2.3 Bootstrap

Definition 5: Bootstrap replication

Let \mathbf{x} be the RV defined as the samples from an arbitrary distribution:

$$\mathbf{x} = \{x_i | 1 \leq i \leq n\}. \quad (70)$$

Let \mathbf{x}^* be the RV defined as the random samples of \mathbf{x} with replacement for each x_i with equal probability, $1/n$:

$$\mathbf{x}^* = \{x_i^* | 1 \leq i \leq n, x_i^* \in \mathbf{x}\}. \quad (71)$$

The Bootstrap replication \mathbf{b} of \bar{x} is defined as the mean of \mathbf{x}^* :

$$\mathbf{b} = \left\{ b_k \mid b_k = \frac{1}{n} \sum_{i=1}^n x_i^*, 1 \leq k \leq nb \right\}, \quad (72)$$

where nb is the number of Bootstrap replications.

Definition 6: Sorted Bootstrap replications

Let the RV \mathbf{B} stands for the sorted Bootstrap replications and defined as:

$$\mathbf{B} = \left\{ B_k \mid B_k \in \mathbf{b}; B_i \leq B_j \text{ if } i < j; 1 \leq i, j, k \leq nb \right\}. \quad (73)$$

Definition 7: Cumulative distribution function (c.d.f.) of \mathbf{B}

The probability that the RV \mathbf{B} is smaller than an arbitrary value b is defined as:

$$GB(b) = P\{\mathbf{B} \leq b\} = \left\{ \frac{k_b}{nb} \mid B_i \leq b, 1 \leq i \leq k_b \right\}. \quad (74)$$

Definition 8: \mathbf{a} -percentile of Bootstrap

The \mathbf{a} -Bootstrap percentile is defined as:

$$\mathbf{q}_a = GB^{-1}(\mathbf{a}), \quad 0 \leq \mathbf{a} \leq 1. \quad (75)$$

Definition 9: Percentile confidence interval of Bootstrap

There are several ways of calculating confidence intervals of the Bootstrap replications. The most straightforward one for the $1-2\mathbf{a}$ Bootstrap confidence interval is the percentile Bootstrap confidence interval, and is defined as the interval that can cover $(1-2\mathbf{a}) * nb$ Bootstrap replications:

$$[\mathbf{q}_{\%,lo}, \mathbf{q}_{\%,up}] = [GB^{-1}(\mathbf{a}), GB^{-1}(1-\mathbf{a})], \quad 0 \leq \mathbf{a} \leq 0.5. \quad (76)$$

Definition 10: Bias-Corrected and accelerated (BCa) confidence interval

The BCa confidence intervals are complicated to describe but are as easy to use as the percentile confidence interval [87]:

$$\begin{aligned} [\mathbf{q}_{BCa,lo}, \mathbf{q}_{BCa,up}] &= [GB^{-1}(\mathbf{a}_1), GB^{-1}(\mathbf{a}_2)] \\ \mathbf{a}_1 &= G\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{\mathbf{a}}}{1 - \hat{a}(\hat{z}_0 + z_{\mathbf{a}})}\right) . \\ \mathbf{a}_2 &= G\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{1-\mathbf{a}}}{1 - \hat{a}(\hat{z}_0 + z_{1-\mathbf{a}})}\right) \end{aligned} \quad (77)$$

The \hat{z}_0 is designated as the Bias-Correction coefficient. It is simply derived from the portion of Bootstrap replications that are smaller than \bar{x} (the sample mean of \mathbf{x}):

$$\hat{z}_0 = G^{-1}\left(\frac{\#\{b < \bar{x}\}}{nb}\right), \quad (78)$$

where the $\#\{b < \bar{x}\}$ is the number of *Bootstrap Replications* that are smaller than \bar{x} . The \hat{a} is designated as the acceleration coefficient. Before defining \hat{a} , the definition of the Jackknife value, $J_{(i)}$, of \bar{x} is defined as:

$$\mathbf{J}_{(i)} = \left\{ J_{(i)} \mid J_{(i)} = \frac{1}{n-1} \left[\left(\sum_{j=1}^n x_j \right) - x_i \right], 1 \leq i \leq n \right\}. \quad (79)$$

The mean of $J_{(i)}$ designate as $J_{(.)}$ is defined as:

$$J_{(.)} = \frac{1}{n} \sum_{j=1}^n J_{(i)}. \quad (80)$$

The acceleration coefficient is then defined as:

$$\hat{a} = \frac{\sum_{i=1}^n (J_{(.)} - J_{(i)})^3}{6 \times \left\{ \sum_{i=1}^n (J_{(.)} - J_{(i)})^2 \right\}^{3/2}}. \quad (81)$$

The BCa Bootstrap estimation of \mathbf{m}_x is defined as the 0.5-percentile of the distribution of BCa Bootstrap replications:

$$\begin{aligned} \bar{x}_{BCa} &= GB^{-1}\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{0.5}}{1 - \hat{a}(\hat{z}_0 + z_{0.5})}\right) \\ &= GB^{-1}\left(\hat{z}_0 + \frac{\hat{z}_0}{1 - \hat{a}\hat{z}_0}\right) \end{aligned} \quad (82)$$

The Bias-Correction coefficient \hat{z}_0 is designed to compensate the difference between \bar{x} and the BCa mean \bar{x}_{BCa} . If the difference between them equals zero, \hat{z}_0 equals zero also. As

for the acceleration coefficient, it refers to the rate of change of the standard error of \bar{x} with respect to \bar{x} , measured on a normalized scale [88]. The larger it is, the wider the confidence interval. Detail discussion about how this acceleration coefficient works is referred to the references⁵. For a normally distributed \bar{x} , the \hat{z}_0 and \hat{a} are both zero, and $\mathbf{a}_1 = \mathbf{a}_2 = \mathbf{a}$. The BCa confidence interval is exactly the same as standard confidence interval.

3.3 Bootstrap Monte Carlo Simulation

3.3.1 Bootstrap Confidence Level

For conventional Monte Carlo, the confidence interval and the *rel_err* are calculated with equation (69), in which the *rel_err* totally relies on the assumption that the distribution of \bar{x} is normal. The possibility that the distribution of \bar{x} might be skewed or platykurtic is ignored. Bootstrap technique, can be used to adjust the confidence level when the normality of the population is poor.

For a given error level, ϵ , the acceptable range for the real mean \mathbf{m}_x is defined as:

$$[A_{lo}, A_{up}] = [(1 - \epsilon) \text{MAX}(\bar{x}, \bar{x}_{BCa}), (1 + \epsilon) \text{MIN}(\bar{x}, \bar{x}_{BCa})]. \quad (83)$$

The acceptable range covers the safe range into which the real mean \mathbf{m}_x can land without violating the user specified error level ϵ , with respect to either \bar{x} or \bar{x}_{BCa} . And then, the Bootstrap confidence level is defined with:

$$\mathbf{a}_{BCa} = GB(A_{lo}) + 1 - GB(A_{up}). \quad (84)$$

3.3.2 Bootstrap Monte Carlo Method

With the \mathbf{a}_{BCa} from equation (84), whether the user specified confidence level is guaranteed or not can be easily determined. The Bootstrap Monte Carlo (BMC) method is demonstrated with the pseudo-codes in Figure 12:

```

Bootstrap Monte Carlo ()
Pwr, e, a; /* Conventional Monte Carlo parameters */
nb; /* Bootstrap parameter */
{
    nSamples=1; rel_err = 1; #Boot =0;
     $z_a = G^{-1}(\mathbf{a})$ ;
    while ( rel_err  $\geq e$  ) {
        nSamples++;
        get new sample  $x_n$  from Pwr;
        update sample mean  $\bar{x}$  and sample variance  $s^2$ ;
        update rel_err;
        if (rel_err  $\leq e$  ) {
            Generate nb Bootstrap Replications b from x;
            Calculate  $\mathbf{a}_{BCa}$  from b;
            #Boot ++;
            if ( $\mathbf{a}_{BCa} > 2 * \mathbf{a}$  ) {
                rel_err = 1;
            }
        }
    }
    return nSamples, nBoot, and  $\bar{x}$ ;
}

```

Figure 12: Pseudo code of BMC

3.3.3 Bootstrap Monte Carlo vs. Conventional Monte Carlo

The proposed BMC method is tested with estimating the average powers of the *ISCAS-85* benchmark circuits. There are 10,000 input vectors in the input sequence for each circuit. The input sequence is a compound of 3 segments: a counter sequence, a LFSR sequence and a sequence of pseudo random numbers. Note that half of the input vectors in the counter sequence have only single input change, and therefore consumes less power. The LFSR sequence represents the input vectors with temporal correlations. The pseudo random input vectors, on the other hand, are spatially and temporally independent. The arrangement of the input sequence is to give the estimator a tough situation because the power histogram of such an input sequence is most likely to be skewed, long tailed and platykurtic at the same time. Besides the *ISCAS-85* benchmarks, an additional circuit, *add_mpr*, is included. It is a circuit with a mode controlling input that controls the function of the circuit to be an adder or a multiplier. The power histogram of it is a typical bimodal distribution. The experimental results are listed in Table 6 to 8.

Table 6: Conventional Monte Carlo vs. Bootstrap Monte Carlo with $\alpha=0.05$

| Circuit | MC | BMC |
|---------|----|-----|
|---------|----|-----|

| | viol_r | nVecs | viol_r | nVecs | #Boot |
|---------|--------|---------|--------|---------|---------|
| C432 | 0.1235 | 1538.51 | 0.0949 | 1786.97 | 14.4926 |
| C499 | 0.1164 | 585.86 | 0.1050 | 629.34 | 2.9860 |
| C880 | 0.1218 | 616.24 | 0.1010 | 663.19 | 3.1880 |
| C1355 | 0.1145 | 633.05 | 0.1005 | 682.92 | 3.2847 |
| C1908 | 0.1194 | 908.35 | 0.1004 | 996.04 | 5.4834 |
| C3540 | 0.1586 | 154.90 | 0.1563 | 156.16 | 1.0405 |
| C6288 | 0.1259 | 1344.78 | 0.1014 | 1532.66 | 11.0338 |
| add_mpr | 0.1384 | 281.35 | 0.1288 | 288.27 | 1.2202 |
| Max | 0.1586 | 1538.51 | 0.1563 | 1786.97 | 14.4926 |
| Avg | 0.1273 | 757.88 | 0.1110 | 841.94 | 5.3412 |

Table 7: Conventional Monte Carlo vs. Bootstrap Monte Carlo with $\alpha=0.025$

| Circuit | MC | | BMC | | |
|---------|--------|---------|--------|---------|---------|
| | viol_r | nVecs | viol_r | nVecs | #Boot |
| C432 | 0.0748 | 2055.63 | 0.0487 | 2504.36 | 25.7884 |
| C499 | 0.0591 | 820.55 | 0.0497 | 886.94 | 4.4027 |
| C880 | 0.0611 | 862.70 | 0.0497 | 935.03 | 4.7865 |
| C1355 | 0.0594 | 884.39 | 0.0499 | 961.19 | 4.9675 |
| C1908 | 0.0653 | 1248.65 | 0.0525 | 1397.90 | 9.0968 |
| C3540 | 0.0832 | 230.54 | 0.0810 | 232.59 | 1.0764 |
| C6288 | 0.0710 | 1812.33 | 0.0485 | 2150.83 | 19.6258 |
| add_mpr | 0.0688 | 408.67 | 0.0627 | 421.19 | 1.5073 |
| Max | 0.0832 | 2055.63 | 0.0810 | 2504.36 | 25.7884 |
| Avg | 0.0678 | 1040.43 | 0.0553 | 1186.25 | 8.9064 |

Table 8: Conventional Monte Carlo vs. Bootstrap Monte Carlo with $\alpha=0.005$

| Circuit | MC | | BMC | | |
|---------|--------|---------|--------|---------|---------|
| | viol_r | nVecs | viol_r | nVecs | #Boot |
| C432 | 0.0266 | 3089.78 | 0.0092 | 4145.60 | 59.5692 |

| | | | | | |
|---------|--------|---------|--------|---------|---------|
| C499 | 0.0163 | 1345.06 | 0.0115 | 1480.03 | 8.3435 |
| C880 | 0.0179 | 1408.56 | 0.0111 | 1558.86 | 9.2150 |
| C1355 | 0.0180 | 1442.10 | 0.0138 | 1600.27 | 9.6379 |
| C1908 | 0.0198 | 1980.65 | 0.0121 | 2322.61 | 19.8780 |
| C3540 | 0.0160 | 409.03 | 0.0146 | 416.39 | 1.3426 |
| C6288 | 0.0238 | 2768.35 | 0.0105 | 3561.91 | 44.9796 |
| add_mpr | 0.0130 | 700.62 | 0.0111 | 728.20 | 2.9168 |
| Max | 0.0266 | 3089.78 | 0.0146 | 4145.6 | 59.5692 |
| Avg | 0.0189 | 1643.02 | 0.0117 | 1976.73 | 19.4853 |

Tables 6 to 8 are the results for \mathbf{a} equals 0.05, 0.025 and 0.005 respectively. The corresponding confidence levels are 90%, 95% and 99%. For each \mathbf{a} , there are results for every circuit with both conventional Monte Carlo method (MC) and the proposed Bootstrap Monte Carlo method (BMC). Each method is performed 10,000 times for each circuit to estimate their average power consumption. The error level \mathbf{e} is set to 0.05. If the error percentage of the estimated power exceeds \mathbf{e} , the number of violations is increased by one. The $viol_r$ columns are the violation ratio defined as the number of violations divided by the number of Monte Carlo simulations:

$$viol_r = \frac{\#\left\{\frac{\bar{x} - m_x}{m_x} > \mathbf{e}\right\}}{10,000} \quad (85)$$

The $nVecs$ columns contain the numbers of input vectors sampled by the corresponding estimation methods. For a good estimation method, the $viol_r$ should be close to and always smaller than $2*\mathbf{a}$. For two estimation methods with the same $viol_r$, the one with smaller $nVecs$ is the better one. There is one additional $\#Boot$ column for BMC. It is the average number of times that Bootstrap process being invoked for a BMC estimation. It is roughly in proportion to the number of additional samples required for BMC. With a larger $\#Boot$, the overhead of using Bootstrap is greater. For the circuits that the $viol_r$ of MC exceed $2*\mathbf{a}$ more, the $\#Boot$ for BMC is supposed to be larger to keep the $viol_r$ of BMC within $2*\mathbf{a}$ safe range. As demonstrated in the tables, the $viol_r$ for conventional Monte Carlo method exceeds $2*\mathbf{a}$ in all 24 cases. On the other hand, with the Bootstrap method monitoring confidence level in

BMC, the $viol_r$ are either under or close to the $2*a$, except for 2 cases: C3540 and add_mpr . One thing needs to be noticed is that the $nVecs$ for C3540 and add_mpr are the two smallest ones among all circuits. If the number of samples is too small to represent the original population, e.g. pre-matured samples, BMC might fail to keep the $viol_r$ within $2*a$ sometimes. That is because the Bootstrap method produces its Bootstrap replications from the samples of Monte Carlo. This drawback will be discussed and eliminated with the proposed Bootstrap Monte Carlo with adaptive stratification method (BMCAS) in the following section. Regardless of this deficiency, the proposed Bootstrap Monte Carlo method is more trustable than conventional Monte Carlo method with about 10% increasing in $nVecs$.

3.4 Adaptive Stratified Random Sampling

Stratification is a technique to divide the sample space into subspaces to reduce the sample variance so that the Monte Carlo can converge sooner with smaller number of samples, n , and achieve better compaction ratio. An indicator function for stratification is a function that returns a value closely related to or even equals the power consumption of input vectors. To build an indicator function, the multiple regression method is adopted.

3.4.1 Single Variable Linear Regression

Given a collection of n data points of two variables x and y :

$$(\mathbf{x}, \mathbf{y}) = \{ (x_i, y_i) | 1 \leq i \leq n \} \quad (86)$$

The best line describing the relation between x and y is defined as:

$$\begin{aligned} \hat{y} &= a\mathbf{x} + b \\ a &= \frac{E[\mathbf{xy}] - E[\mathbf{x}]E[\mathbf{y}]}{E[\mathbf{xx}] - E[\mathbf{x}]E[\mathbf{x}]} \\ b &= E[\mathbf{y}] - aE[\mathbf{x}] \end{aligned} \quad (87)$$

where $E[\]$ is the function of expected value, and \hat{y} is the predictor of y . This predictor is the one with zero bias and minimum RMS error.

3.4.2 Multiple Regression

Multiple regression is simply an extension of the single variable linear regression. Given n data points of $m+1$ variables:

$$\begin{aligned} (\mathbf{X}, \mathbf{y}) &= \{ [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m], \mathbf{y} \} \\ &= \{ \{ [x_{1,i}, x_{2,i}, \dots, x_{m,i}], y_i \} | 1 \leq i \leq n \}. \end{aligned} \quad (88)$$

Note that \mathbf{X} is a row matrix stands for the \mathbf{x} variables. The best function describing the relation between \mathbf{X} and \mathbf{y} is defined as:

$$\begin{aligned}
 \hat{\mathbf{y}} &= \mathbf{X}\mathbf{A} + b \\
 \mathbf{C} &= E[\mathbf{X}\mathbf{y}] - E[\mathbf{X}]E[\mathbf{y}] \\
 \mathbf{D} &= E[\mathbf{X}^T \mathbf{X}] - E[\mathbf{X}]^T E[\mathbf{X}] \\
 \mathbf{A} &= \mathbf{D}^{-1} \mathbf{C}^T \\
 b &= E[\mathbf{y}] - E[\mathbf{X}]\mathbf{A}
 \end{aligned} \tag{89}$$

where \mathbf{A} is a column matrix that contains the coefficients for each $\mathbf{x}_1, \dots, \mathbf{x}_m$. Note that the order of \mathbf{X} and \mathbf{A} are switched to make the product of them a scalar.

3.4.3 Variable Selection

As we can see in equation (89), deriving the coefficient matrix \mathbf{A} includes a matrix inverse operation on matrix \mathbf{D} . If matrix \mathbf{D} were singular, the equation of measuring the coefficient matrix \mathbf{A} would fail to be solved. To prevent this, the basic assumptions of multiple regression need to be taken into consideration while selecting the \mathbf{x} variables.

Assumption 1: The relation between \mathbf{y} and \mathbf{x}_i is linear.

The \mathbf{y} variable in power estimation is the power consumption of each input vector. The power consumption of CMOS circuits is dominated by the dynamic power. The dynamic power is consumed when the inputs of the circuits are switching. In other words, more input switching implies that more dynamic power is consumed. This makes the switching conditions of the primary inputs good candidates for the \mathbf{x} variables.

Assumption 2: The \mathbf{x}_i variables are mutually independent.

For a primary input, there are 4 possible transitions between 2 consecutive clock cycles: 0 to 0, 0 to 1, 1 to 0, and 1 to 1.

Let the Boolean value for the i^{th} input in the j^{th} clock cycle be denoted as b_i^j , the input pattern in the j^{th} clock cycle is defined as:

$$\text{Pat}^j = \{ b_i^j \mid b_i^j \in \{0,1\}, 1 \leq i \leq n_in, 0 \leq j \leq N \} \tag{90}$$

where n_in stands for the number of primary inputs, and N stands for the total number of input vectors. The j^{th} input vector is defined as two consecutive input patterns:

$$V^j = \left\{ (Pat^j, Pat^{j+1}) \mid 0 \leq j \leq N-1 \right\} \quad (91)$$

There are four transition variables designated for the transition behavior of the each input:

$$\begin{aligned} & \mathbf{T}(i, V^j) \\ &= \left\{ T_k(i, V^j) \mid 0 \leq k \leq 3, 1 \leq i \leq n_in, 0 \leq j \leq N-1 \right\}, \end{aligned} \quad (92)$$

$$\text{where } T_k(i, V^j) = \begin{cases} 1, & \text{if } k = 2 \times b_i^j + b_i^{j+1} \\ 0, & \text{otherwise} \end{cases}.$$

For an input i and input vector V^j , one and only one of the four transitions can take place.

This makes the four transition variables mutually dependent:

$$T_0(i, V^j) + T_1(i, V^j) + T_2(i, V^j) + T_3(i, V^j) = 1 \quad (93)$$

Therefore, at most three out of the four transition variables need to be chosen as the \mathbf{x} variables for multiple regressions. We select the T_0 , T_1 and T_2 because they are smaller and continuous by the index. Note that, any combination of three transition variables chosen is not losing generality because the removed one can always be derived from the others selected.

After choosing the \mathbf{x} variables for equation (88), it can be rewritten in the form of constructing the predictor for the power consumption of each input vector. Let the power consumption of each input vector indicated by the indicator function be:

$$\begin{aligned} \hat{Power}(\mathbf{V}) &= \mathbf{X}(\mathbf{V})\mathbf{A} + b \\ \mathbf{X}(\mathbf{V}) &= \{ [T_0(i, \mathbf{V}), T_1(i, \mathbf{V}), T_2(i, \mathbf{V})] \mid 1 \leq i \leq n_in \}, \end{aligned} \quad (94)$$

where \mathbf{V} is the set of all input vectors. In the same manner, the coefficient matrix \mathbf{A} can be derived from the multiple regression equations:

$$\begin{aligned} \mathbf{C} &= E[\mathbf{X}(\mathbf{W})P_{wr}(\mathbf{W})] - E[\mathbf{X}(\mathbf{W})]E[P_{wr}(\mathbf{W})] \\ \mathbf{D} &= E[\mathbf{X}(\mathbf{W})^T \mathbf{X}(\mathbf{W})] - E[\mathbf{X}(\mathbf{W})]^T E[\mathbf{X}(\mathbf{W})] \\ \mathbf{A} &= \mathbf{D}^{-1} \mathbf{C}^T \\ b &= E[P_{wr}(\mathbf{W})] - E[\mathbf{X}(\mathbf{W})]\mathbf{A} \end{aligned} \quad (95)$$

where \mathbf{W} is the set of sampled input vectors and $P_{wr}(\mathbf{W})$ are their corresponding power consumption measured with simulator.

3.4.4 BMC with Adaptive Stratification (BMCAS)

With the coefficient matrix \mathbf{A} and equation (94), the population can be stratified into a certain number of strata. Initially, the population is not stratified, and the first few samples x_i 's are randomly sampled. After the number of sampled input vectors is larger than a predefined

threshold, the multiple regression function is invoked to derive A . The $\hat{Power}(\mathbf{V})$ is calculated with equation (94) and the stratification process starts. Hence, the stratified random sampling process takes over the place of random sampling. After some other new input vectors are sampled, the multiple regression process is executed again to recalculate a better A for re-stratification. The pseudo code for the proposed Bootstrap Monte Carlo with Adaptive Stratification (BMCAS) is depicted in Figure 13.



```

Bootstrap Monte Carlo with Adaptive Stratification ()
Pwr, e, a; /* Conventional Monte Carlo parameters */
nb; /* Bootstrap parameter */
{
  nSamples=0; rel_err = 1; nRestrat =0; keep_sampling = 0;
  W = {∅}; S = {∅}; V = {Vi | 1≤i≤N}
  za = G-1(a);
  while ( rel_err ≥ e ) {
    nSamples++;
    if (stratified)
      Sample input vectors v from V with stratified sampling;
    else
      Sampled input vectors v from V with random sampling;
    S = S ∪ v; W = W ∪ v;
    Get y = Power(v); from simulator;
    y = y ∪ y;
    if (keep_sampling) {
      keep_sampling--; rel_err = 1; continue;
    } else {
      update x̄, s2, and rel_err;
    }
    if (rel_err ≥ e) {
      if (#(W) > 9 * n_in) {
MR:      X = T(S);
          if (A = Multiple_Regression(X,y) is success) {
            Power(V) = T(V)A + b;
            Reostratification(V);
            stratified = TRUE; W = {∅}; nRestrat++;
          }
        }
      } else {
        Generate nb Bootstrap Replications b from x;
        Calculate aBCa from b;
        #Boot ++;
        if (aBCa > 2 * a) {
          rel_err = 1;
        } else if (!stratified) {
          rel_err = 1; keep_sampling = 2; goto MR;
        } esle {
          return nSamples, nRestrat, and x̄;
        }
      }
    }
  }
}

```

Figure 13: Pseudo code of BMCAS

The reason of starting multiple regression after the number of W is larger than $9*n_{in}$ is to prevent too many empty elements in matrix D which might lead to a singular matrix for the matrix inverse operation. The *keep_sampling* variable is an insurance to prevent the Bootstrap Monte Carlo failure caused by the premature samples as discussed in previous section. The *rel_err* and the *keep_sampling* are given one and two, respectively when BMCAS exits before

any stratification is performed. With these setting, the BMCAS will stratify the population at least once and samples three new samples after the forced stratification to get at least three samples that are sampled from the population more uniformly.

3.5 Experimental Results

To demonstrate the performance of proposed BMCAS, one more stratification method called Hamming distance method (HDM) is implemented for comparison. It is a stratification method based on the assumptions that power consumption increases with the number of inputs transitions becoming larger. HDM method stratifies the input vectors into strata according to the Hamming distance of each input vector. For both methods, the input vectors are stratified into six equal sized strata. The reason is that one can get little sample variance reduction by setting the number of strata larger than six [19].

The results are in Table 9 to 11. The results from BMC method in section 3.3 are listed in the tables, too. It is designated as *NO_STRAT* because there is no stratification procedure in BMC. Similar to Table 6 to 8, Table 9 to 11 show the results for confidence level 90%, 95% and 99%, respectively. The first columns of them are the names of the circuits. The numbers in the brackets next to the circuit names are the number of inputs of the corresponding circuit. They are listed as a reference because BMCAS re-stratifies the population after $9 \cdot n_{in}$ new input vectors sampled. There are three columns of data for each stratification method. The *nSample* column shows the average number of samples required for the corresponding stratification method to converge to a value of estimated power. The *nVecs* columns are the average numbers of the sampled input vectors. The smaller is the *nVecs*, the better the stratification method is. The error level ϵ is set to 0.05. The *viol_r* column shows the violation ratios with the same definition as equation (85), and the *#Boot* columns are of the same definition as in tables 6 to 8. The extra *#ReStrat* column for BMCAS is the average number of stratification being performed.

Comparing the proposed BMCAS and the BMC, the *viol_r* of BMCAS are closer to $2 \cdot \mathbf{a}$ than BMC in almost all circuits and all confidence level. Besides, with the proposed adaptive stratification technique, the numbers of sampled input vectors are about 27% smaller than those of BMC in average. For some circuits, BMCAS even needs only half of the number of input vectors that BMC needs. The *#ReStrat* for some circuits, like C3540, are equal to 1.0

exactly because all of the BMC estimations exit with a number of sampled input vectors smaller than the threshold for starting stratification, and BMCAS will perform at least one stratification process before exiting. As shown in tables 9 to 11, the $viol_r$ for C3540 are safely kept within $2*\mathbf{a}$. As for the results from HDM method, although its $nVecs$ is the smallest, the violation ratio exceeds $2*\mathbf{a}$ for all cases. This makes the results from HDM un-trustable and the $nVecs$ meaningless. With the safely kept $viol_r$ and the smaller number of $nVecs$, we can summarize that the proposed BMCAS is the most reliable one and can efficient reduce the required number approach.

Table 9: Comparison of stratification method, confidence level = 90%

| Circuit | NO_STRAT | | | HDM | | | BMCAS | | | | |
|---------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|-----------|-------------|----------|
| | $nSample$ | $nVecs$ | $viol_r$ | $nSample$ | $nVecs$ | $viol_r$ | $nSample$ | $nVecs$ | $viol_r$ | $\#ReStrat$ | $\#Boot$ |
| C432 | 297.74 | 1786.47 | 0.0949 | 48.56 | 291.36 | 0.1596 | 193.99 | 1163.97 | 0.0918 | 3.6702 | 7.8359 |
| C499 | 104.89 | 629.34 | 0.1050 | 4.96 | 29.79 | 0.1421 | 90.72 | 544.30 | 0.0739 | 1.9919 | 3.8312 |
| C880 | 110.53 | 663.19 | 0.1010 | 13.02 | 78.12 | 0.2141 | 107.92 | 647.56 | 0.0870 | 1.8311 | 3.7967 |
| C1355 | 113.82 | 682.92 | 0.1005 | 4.85 | 29.10 | 0.1388 | 94.46 | 566.80 | 0.0770 | 1.9974 | 3.9666 |
| C1908 | 166.00 | 996.04 | 0.1004 | 10.60 | 63.59 | 0.2060 | 105.05 | 630.31 | 0.0795 | 2.2278 | 2.2278 |
| C3540 | 26.03 | 156.16 | 0.1563 | 7.10 | 42.61 | 0.1923 | 30.81 | 184.89 | 0.0982 | 1.0000 | 2.0777 |
| C6288 | 255.44 | 1532.66 | 0.1014 | 10.36 | 62.16 | 0.1969 | 138.80 | 832.80 | 0.0831 | 3.0274 | 5.2061 |
| add_mpr | 48.04 | 288.27 | 0.1288 | 43.35 | 260.07 | 0.1628 | 53.22 | 319.32 | 0.0911 | 1.3348 | 2.4049 |
| Max | 297.74 | 1786.47 | 0.1563 | 48.56 | 291.36 | 0.2141 | 193.99 | 1163.97 | 0.0982 | 3.6702 | 7.8359 |
| Avg | 140.31 | 841.88 | 0.1110 | 17.85 | 107.1 | 0.1766 | 101.87 | 611.24 | 0.0852 | 2.1351 | 3.9184 |

Table 10: Comparison of stratification method, confidence level = 95%

| Circuit | NO_STRAT | | | HDM | | | BMCAS | | | | |
|---------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|-----------|-------------|----------|
| | $nSample$ | $nVecs$ | $viol_r$ | $nSample$ | $nVecs$ | $viol_r$ | $nSample$ | $nVecs$ | $viol_r$ | $\#ReStrat$ | $\#Boot$ |
| C432 | 417.39 | 2504.36 | 0.0487 | 72.63 | 435.80 | 0.0703 | 261.99 | 1571.98 | 0.0393 | 4.6465 | 12.3182 |
| C499 | 147.82 | 886.94 | 0.0497 | 6.96 | 41.74 | 0.0852 | 108.05 | 648.32 | 0.0392 | 2.0039 | 3.8246 |
| C880 | 155.84 | 935.03 | 0.0497 | 20.14 | 120.83 | 0.1193 | 132.88 | 797.26 | 0.0458 | 1.9994 | 4.2675 |
| C1355 | 160.20 | 961.19 | 0.0499 | 6.77 | 40.66 | 0.0847 | 111.59 | 669.55 | 0.0400 | 2.0152 | 3.8161 |

| | | | | | | | | | | | |
|---------|--------|---------|--------|-------|--------|--------|--------|---------|--------|--------|---------|
| C1908 | 232.98 | 1397.90 | 0.0525 | 16.29 | 97.75 | 0.1148 | 131.98 | 791.87 | 0.0430 | 2.9044 | 4.7751 |
| C3540 | 38.77 | 232.59 | 0.0810 | 10.42 | 62.51 | 0.1158 | 43.19 | 259.15 | 0.0457 | 1.0000 | 2.1437 |
| C6288 | 358.47 | 2150.83 | 0.0485 | 15.91 | 95.45 | 0.1146 | 188.89 | 1133.32 | 0.0395 | 3.9200 | 7.7984 |
| add_mpr | 70.20 | 421.19 | 0.0627 | 64.84 | 389.06 | 0.0768 | 68.21 | 409.27 | 0.0443 | 1.9117 | 2.7532 |
| Max | 417.39 | 2504.36 | 0.0810 | 72.63 | 435.80 | 0.1193 | 261.99 | 1571.98 | 0.0458 | 4.6465 | 12.3182 |
| Avg | 197.71 | 1186.25 | 0.0553 | 26.75 | 160.48 | 0.0977 | 130.85 | 785.09 | 0.0421 | 2.5501 | 5.2121 |

Table 11: Comparison of stratification method, confidence level = 99%

| Circuit | NO_STRAT | | | HDM | | | BMCAS | | | | |
|---------|----------------|--------------|---------------|----------------|--------------|---------------|----------------|--------------|---------------|-----------------|--------------|
| | <i>nSample</i> | <i>nVecs</i> | <i>viol_r</i> | <i>nSample</i> | <i>nVecs</i> | <i>viol_r</i> | <i>nSample</i> | <i>nVecs</i> | <i>viol_r</i> | <i>#ReStrat</i> | <i>#Boot</i> |
| C432 | 690.93 | 4145.60 | 0.0092 | 125.95 | 755.71 | 0.0135 | 425.68 | 2554.10 | 0.0073 | 6.8489 | 26.3906 |
| C499 | 246.67 | 1480.03 | 0.0115 | 11.88 | 71.27 | 0.0269 | 148.93 | 893.58 | 0.0097 | 2.7605 | 4.8357 |
| C880 | 259.80 | 1558.86 | 0.0111 | 37.33 | 223.95 | 0.0316 | 186.26 | 1117.56 | 0.0120 | 2.1826 | 5.9437 |
| C1355 | 266.71 | 1600.27 | 0.0138 | 11.53 | 69.20 | 0.0275 | 155.74 | 934.48 | 0.0093 | 2.8516 | 5.1525 |
| C1908 | 387.10 | 2322.61 | 0.0121 | 30.19 | 181.13 | 0.0330 | 209.86 | 1259.15 | 0.0078 | 4.2333 | 8.1702 |
| C3540 | 69.34 | 416.39 | 0.0146 | 18.92 | 113.53 | 0.0367 | 73.16 | 438.95 | 0.0094 | 1.2434 | 2.5202 |
| C6288 | 593.65 | 3561.91 | 0.0105 | 29.90 | 179.37 | 0.0325 | 315.67 | 1894.00 | 0.0093 | 6.0004 | 16.0834 |
| add_mpr | 123.03 | 728.20 | 0.0111 | 112.04 | 672.25 | 0.0137 | 93.23 | 559.35 | 0.0074 | 2.0002 | 3.9304 |
| Max | 690.93 | 4145.60 | 0.0146 | 125.95 | 755.71 | 0.0367 | 425.68 | 2554.10 | 0.0120 | 6.8489 | 26.3906 |
| Avg | 329.65 | 1976.73 | 0.0117 | 47.22 | 283.30 | 0.0269 | 201.07 | 1206.40 | 0.0090 | 3.5151 | 9.1283 |

3.6 Summary

The Bootstrap technique is adopted in this chapter to assure the confidence level when doing Monte Carlo estimation. With this technique, the Monte Carlo method can be improved since confidence level is monitored during the Monte Carlo simulation. Besides, we proposed a novel adaptive stratification technique, with which the population can be dynamically and well stratified to keep the sample variance minimized. The experimental results on the *ISCAS-85* benchmarks show that the proposed Bootstrap Monte Carlo with adaptive

stratification (BMCAS) successfully preserves the confidence level while efficiently reducing the required number of input vectors.



4 Automatic Power Profiler

Estimating average power is a passive approach for simulator to demonstrate the power consumption of a circuit. It relies on what input sequence users gave to the simulator to estimate power with. Most of the time, even the designers of the circuit under test are not aware of how the power consumption varies with different input sequences. Power profiler is such a product that provides users an easy way to visually examine the distribution of the power consumption.

The easiest way to get the profile of the power consumption is to simulate all possible input vectors and plot the histogram of the simulated power. However, this is not practical for large circuits or circuits with a large number of inputs. Besides, more information about how the power consumption varies with the input logic values or input transitions is necessary for optimizing the power consumption. In the following sections, the proposed measurement for stratifying unlimited length input vectors will be defined. The implemented tool named PowerPro that integrates the stratification scheme and the SPICE simulator will also be detailed. With PowerPro, users can not only easily get the plot of power consumption distribution for examining the average power consumption, but also get the worst case input combination that tends to induce worst case power consumption.

4.1 Problem Definition

Given a circuit and the input statistics, the objective of power profiling is to efficiently and accurately plot the distribution of power consumption of a circuit. With efficiently, that means the task should be done with the minimum number of input vectors fed into the simulator. With accurately, that means the input vectors are stratified into strata and the mean of each stratum should be correctly estimated before plotting the distribution. For example, if the histogram of the power consumption of a circuit is as shown in Figure 14 (a):

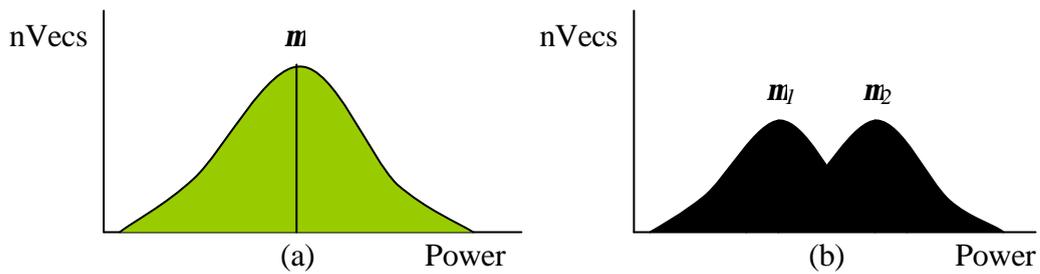


Figure 14: Example of power distribution

It is strongly demanded to easily stratify the population into two or more sub-populations, as shown in Figure 14 (b), if they have smaller number of input vectors needed to be profiled.

Stratification for unlimited length input sequence requires a clear and logical way of deciding which stratum should an input vector belongs to. The BMCAS method proposed in Chapter 3 that stratifies the input vectors according to the order of the input vectors sorted with the sum of power sensitivities is not practical because the number of input vectors can be too long for any sorting algorithm. Hamming distance method is a primitive and feasible way of stratifying the input sequence because it only needs the number of inputs that are toggling for stratification. However, there are two inherent problems with the Hamming distance method. The first problem of Hamming distance method is that it treats the toggling of all inputs with the same importance. In fact, the power consumptions induced by the toggling of different inputs are usually different due to the difference of the sizes of their fan out cone. Regarding this problem, the power sensitivities are utilized for analyzing the number of samples needed so that a better way of stratification that requires less number of samples can be achieved.

Another shortcoming of Hamming distance method is that there are four possible transitions for an input to make between two consecutive clock cycles and each of them may induce high power consumption. For example, a D flip-flop that is triggered by the positive edge of the clock input, consumes the largest power when the clock input is making a positive edge transition. When the clock input is making a 1 to 0 transition, the power consumption will be much smaller. For memory circuits, the enable signal is responsible for enabling or disabling the memory with logic level 0 or 1. In this case, neither the 0 to 1 nor the 1 to 0 transition induces the power consumption. Instead, the power consumption highly depends on the logic value of the enable signal. To solve this problem, a new method of finding the transitions for each input that may induce high power consumption that are called power sensitive transition

(POST) is proposed. The definition and the derivation of the POST will be detailed in the following sections.

4.2 Number of Samples

For the sub-populations in Figure 14 (b), let the variances be \mathbf{s}_1^2 and \mathbf{s}_2^2 , the means be \mathbf{m}_1 and \mathbf{m}_2 and the numbers of input vectors be n_1 and n_2 , the mean and the variance for the population in Figure 14 (a) can be derived as

$$\begin{aligned}\mathbf{m} &= \frac{n_1 \mathbf{m}_1 + n_2 \mathbf{m}_2}{n_1 + n_2}, \\ \mathbf{s}^2 &= \frac{n_1 \mathbf{s}_1^2 + n_2 \mathbf{s}_2^2}{(n_1 + n_2)} + \left(\frac{n_1 \mathbf{m}_1 - n_2 \mathbf{m}_2}{n_1 + n_2} \right)^2.\end{aligned}\quad (96)$$

The expected number of samples to get a convergence in the Monte Carlo criterion for Figure 14 (a) is:

$$\left(\frac{z_{1-a} \mathbf{s}}{err \times \mathbf{m}} \right)^2. \quad (97)$$

On the other hand, the required numbers of samples to get the criterion met for the two separated sub-populations are:

$$\left(\frac{z_{1-a} \mathbf{s}_1}{err \times \mathbf{m}_1} \right)^2, \quad (98)$$

and

$$\left(\frac{z_{1-a} \mathbf{s}_2}{err \times \mathbf{m}_2} \right)^2. \quad (99)$$

To be benefited by dividing the population, the following criterion must be satisfied:

$$\left(\frac{z_{1-a} \mathbf{s}_1}{err \times \mathbf{m}_1} \right)^2 + \left(\frac{z_{1-a} \mathbf{s}_2}{err \times \mathbf{m}_2} \right)^2 < \left(\frac{z_{1-a} \mathbf{s}}{err \times \mathbf{m}} \right)^2. \quad (100)$$

By dividing both side with z_{1-a} and err ,

$$\left(\frac{\mathbf{s}_1}{\mathbf{m}_1} \right)^2 + \left(\frac{\mathbf{s}_2}{\mathbf{m}_2} \right)^2 < \left(\frac{\mathbf{s}}{\mathbf{m}} \right)^2. \quad (101)$$

Substituting (96) into (101),

$$\left(\frac{\mathbf{s}_1}{\mathbf{m}_1} \right)^2 + \left(\frac{\mathbf{s}_2}{\mathbf{m}_2} \right)^2 < \frac{(n_1 + n_2)(n_1 \mathbf{s}_1^2 + n_2 \mathbf{s}_2^2)}{(n_1 \mathbf{m}_1 + n_2 \mathbf{m}_2)^2} + \frac{(n_1 \mathbf{m}_1 - n_2 \mathbf{m}_2)^2}{(n_1 \mathbf{m}_1 + n_2 \mathbf{m}_2)^2}. \quad (102)$$

This is the criterion to be benefited from separating the populations. However, there is no information about the mean and the variance of the two populations before simulating all input vectors contained in them. Fortunately, equation (94) provides us an estimator for the power consumption of input vectors with the power sensitivity sum. With equation (94), the estimated means and variances with the power sensitivities measured from the sampled input vectors can be derived as

$$\hat{\mathbf{m}}_1 = E[\hat{P}_{power}(\mathbf{V}_1)], \quad (103)$$

$$\hat{\mathbf{s}}_1^2 = E[\hat{P}_{power}(\mathbf{V}_2)^2] - \hat{\mathbf{m}}_1^2, \quad (104)$$

$$\hat{\mathbf{m}}_2 = E[\hat{P}_{power}(\mathbf{V}_2)], \quad (105)$$

and
$$\hat{\mathbf{s}}_2^2 = E[\hat{P}_{power}(\mathbf{V}_2)^2] - \hat{\mathbf{m}}_2^2. \quad (106)$$

The \mathbf{V}_1 and \mathbf{V}_2 are the input vectors set for sub-population 1 and sub-population 2, respectively.

4.3 POST - Power Sensitive Transition

Power sensitive transition is originated from the knowledge of previous subsections that the power consumption induced by different inputs should have different weighting. Summarizing the above observations, a new method of categorizing an input vector with its power sensitive transition (POST) is defined. The transitions of an input that makes the circuit consume more energy in average are called the POST of the input. The other transitions that are not POST are designated as Non-POST.

Table 12: POST and non-POST combinations

| Case | POST | Non-POST | POST Description |
|-------|----------|----------|-------------------|
| 0/123 | 00 | 01,10,11 | Static-0 |
| 1/023 | 01 | 00,10,11 | Positive Edge |
| 2/013 | 10 | 00,01,11 | Negative Edge |
| 3/012 | 11 | 00,01,10 | Static-1 |
| 012/3 | 00,01,10 | 11 | Non-Static-1 |
| 013/2 | 00,01,11 | 10 | Non-Negative Edge |
| 023/1 | 00,10,11 | 01 | Non-Positive Edge |
| 123/0 | 01,10,11 | 00 | Non-Static-0 |

| | | | |
|-------|-------|-------|-------------------|
| 01/23 | 00,01 | 10,11 | Previous Logic -0 |
| 02/13 | 00,10 | 01,11 | Current Logic -0 |
| 03/12 | 00,11 | 10,01 | Non-Transition |
| 12/03 | 01,10 | 00,11 | Transition |
| 13/02 | 01,11 | 00,10 | Current Logic -1 |
| 23/01 | 10,11 | 00,01 | Previous Logic -1 |

The possible POST/Non-POST combinations are listed in Table 12. The 0,1,2,3 in the ‘‘Case’’ column stands for the 0 to 0, 0 to 1, 1 to 0 and 1 to 1 transitions, respectively. The first 0/123 case is the one that the POST is 0 to 0 transition and the non-POST are 0 to 1, 1 to 0 and 1 to 1 transitions. It is suitable for the kind of input that induces the most power consumption when it is making a 0 to 0 transition. For a low-enabled signal, the most possible POST/non-POST combination for it is the 02/13 because the logic-0 of current clock cycle will enable the circuit and consume power. For a clock input that triggers the circuit to work with rising edge, the most possible POST/non-POST combination for it is 1/023.

Since the purpose of choosing the POST is to reduce the number of sampled vectors required for getting a convergent population mean estimation of each subset, the best POST/Non-POST combination for an input is the one that has the smallest sample variance.

Given an input vector, the sum of the power sensitivities corresponding to the transitions of each input in the input vector is a good indicator for the power consumption, provided that the power sensitivities are properly measured. Let the power sensitivities of the i^{th} input be $\{A_{i0}, A_{i1}, A_{i2}, A_{i3}\}$ and the transition probabilities be $\{P_0, P_1, P_2, P_3\}$ corresponding to the four transitions $\{00, 01, 10, 11\}$. Categorizing the input vectors into four subsets according to the transition condition of the i^{th} input forms four subsets of input vectors $\{\mathbf{V}_{i0}, \mathbf{V}_{i1}, \mathbf{V}_{i2}, \mathbf{V}_{i3}\}$. From equation (94), the power consumption indicator function of the \mathbf{V}_{ik} is defined as:

$$\hat{Power}(\mathbf{V}_{ik}) = A_{ik} + \hat{Power}_i(\mathbf{V}_{ik}), \quad (107)$$

where

$$\hat{Power}_i(\mathbf{V}_{ik}) = \mathbf{X}(\mathbf{V}_{ik})\mathbf{A} + b$$

$$\mathbf{X}(\mathbf{V}_{ik}) = \{[T_0(j, \mathbf{V}_{ik}), T_1(j, \mathbf{V}_{ik}), T_2(j, \mathbf{V}_{ik})] | 1 \leq i \leq n_{in}, j \neq i\}. \quad (108)$$

By assuming the i^{th} input being independent of the other inputs, the sample mean and sample variance of subset \mathbf{V}_{ik} are defined as:

$$\begin{aligned}\mathbf{m}_{ik} &= A_{ik} + \mathbf{m}_{iik} \\ \mathbf{s}_{ik}^2 &= \mathbf{s}_{iik}^2\end{aligned}, \quad (109)$$

where

$$\begin{aligned}\mathbf{m}_{iik} &= \text{mean}(\hat{P}\text{ower}_i(\mathbf{V}_{ik})) \\ \mathbf{s}_{iik}^2 &= \text{variance}(\hat{P}\text{ower}_i(\mathbf{V}_{ik}))\end{aligned} \quad (110)$$

Sampling from the four subsets separately results in the smallest sampling variance, which equals:

$$\mathbf{s}_i^2 = \sum_{k=0}^3 P_k \mathbf{s}_{iik}^2. \quad (111)$$

However, by dividing the input vectors into four subsets according to the transition of each input each subset contain only 1 input vector. In other words, to get a sample of each subset is equivalent to simulating all input vectors and this violates the goal of reducing the number of simulated vectors. Therefore, we abandon this option of dividing the input vectors into four subsets was abandoned. In stead, according to the POST and non-POST, the input vectors can be divided into two subsets only. Take case 0/123 in Table 12 as an example. Dividing the input vectors with 0/123 POST/non-POST combination is to put the \mathbf{V}_{i0} alone and group the $\mathbf{V}_{i1}, \mathbf{V}_{i2}, \mathbf{V}_{i3}$ together. The sampling variance from the POST and the non-POST subset is:

$$\mathbf{s}_i^2 = P_0 \mathbf{s}_{i0}^2 + (P_1 + P_2 + P_3) \mathbf{s}_{ii123}^2 \quad (112)$$

$$\mathbf{s}_{ii123}^2 = \frac{P_1(\mathbf{s}_{ii1}^2 + \mathbf{m}_{i1}^2) + P_2(\mathbf{s}_{ii2}^2 + \mathbf{m}_{i2}^2) + P_3(\mathbf{s}_{ii3}^2 + \mathbf{m}_{i3}^2)}{(P_1 + P_2 + P_3)} \quad (113)$$

where

$$\left(\frac{P_1 \mathbf{m}_{i1} + P_2 \mathbf{m}_{i2} + P_3 \mathbf{m}_{i3}}{(P_1 + P_2 + P_3)} \right)^2.$$

Subtracting equation (112) with equation (111), the difference of the sample variance between dividing the input vectors with the 0/123 POST combination and dividing them into four subsets is expressed as

$$\Delta \mathbf{s}_{i0/123}^2 = P_1(\mathbf{m}_{i1}^2) + P_2(\mathbf{m}_{i2}^2) + P_3(\mathbf{m}_{i3}^2) - \frac{(P_1 \mathbf{m}_{i1} + P_2 \mathbf{m}_{i2} + P_3 \mathbf{m}_{i3})^2}{(P_1 + P_2 + P_3)}. \quad (114)$$

To get a clearer look at equation (113), equation (107) is substituted into it, and the $\mathbf{m}_{i0}, \mathbf{m}_{i1}, \mathbf{m}_{i2}, \mathbf{m}_{i3}$ are assumed to be equal to c to further simplify the result, and then:

$$\Delta \mathbf{s}_{i0/123}^2 = P_1 A_{i1}^2 + P_2 A_{i2}^2 + P_3 A_{i3}^2 - \frac{(P_1 A_{i1} + P_2 A_{i2} + P_3 A_{i3})^2}{(P_1 + P_2 + P_3)}. \quad (115)$$

It is very clear that the assumptions, $\Delta S_{i0/123}^2$ becomes a function of the power sensitivities of the i^{th} input and the transition probabilities of it. The sample variance difference for another POST/non-POST combination 01/23 can be derived similarly:

$$\Delta S_{i01/23}^2 = P_0 A_{i0}^2 + P_1 A_{i1}^2 - \frac{(P_0 A_{i0} + P_1 A_{i1})^2}{(P_0 + P_1)} + P_2 A_{i2}^2 + P_3 A_{i3}^2 - \frac{(P_2 A_{i2} + P_3 A_{i3})^2}{(P_2 + P_3)}. \quad (116)$$

For all POST/Non-POST combinations, the one with the smallest sample variance difference is the one that has the smallest sample variance if the input vectors are divided according to it. Here is an example of POST identification for a circuit with four inputs whose power sensitivities and the transition probabilities are listed in Table 13.

Table 13: POST identification example

| | 00 | | 01 | | 10 | | 11 | |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|
| | A_0 | P_0 | A_1 | P_1 | A_2 | P_2 | A_3 | P_3 |
| In1 | 30 | 0.25 | 100 | 0.2 | 100 | 0.2 | 100 | 0.35 |
| In2 | 10 | 0.15 | 20 | 0.4 | 20 | 0.4 | 10 | 0.05 |
| In3 | 20 | 0.2 | 30 | 0.3 | 30 | 0.3 | 10 | 0.2 |
| In4 | 10 | 0.25 | 150 | 0.25 | 60 | 0.25 | 40 | 0.25 |

Checking the sampling difference of each case in Table 12, the sample variance differences are listed in the following table:

Table 14: Sample variance differences derived from Table 13

| Case | POST | non-POST | In1 | In2 | In3 | In4 |
|-------|----------|----------|-------|--------|-------|--------|
| 0/123 | 00 | 01,10,11 | 0 | 117.6 | 60 | 1716.7 |
| 1/023 | 01 | 00,10,11 | 842.2 | 244.6 | 48.57 | 316.7 |
| 2/013 | 10 | 00,01,11 | 842.2 | 244.6 | 48.57 | 2716.7 |
| 3/012 | 11 | 00,01,10 | 753.9 | 202.1 | 15 | 2516.7 |
| 012/3 | 00,01,10 | 11 | 753.9 | 202.1 | 15 | 2516.7 |
| 013/2 | 00,01,11 | 10 | 842.2 | 244.6 | 48.57 | 2716.7 |
| 023/1 | 00,10,11 | 01 | 842.2 | 244.6 | 48.57 | 316.7 |
| 123/0 | 01,10,11 | 00 | 0 | 117.6 | 60 | 1716.7 |
| 01/23 | 00,01 | 10,11 | 544.4 | 285.65 | 60 | 2500.0 |
| 02/13 | 00,10 | 01,11 | 544.4 | 285.65 | 60 | 1825.0 |

| | | | | | | |
|-------|-------|-------|-------|--------|----|--------|
| 03/12 | 00,11 | 10,01 | 714.6 | 3.75 | 10 | 1125.0 |
| 12/03 | 01,10 | 00,11 | 714.6 | 3.75 | 10 | 1125.0 |
| 13/02 | 01,11 | 00,10 | 544.4 | 285.65 | 60 | 1825.0 |
| 23/01 | 10,11 | 00,01 | 544.4 | 285.65 | 60 | 2500.0 |

For In1, although the 0/123 and 123/0 combinations both have the smallest sample variance difference, the 123/0 is chosen because the average power consumption is higher when In1 is making a 0 to 1, 1 to 0 or 1 to 1 transition. Same rule is used for picking the POST/non-POST combination 12/03 for In2 and 1/023 for In4.

4.4 Stratification with POST

With the POST of an input, say In1, the original population can be stratified into two sub-populations according to the condition that the input is making a POST or not:

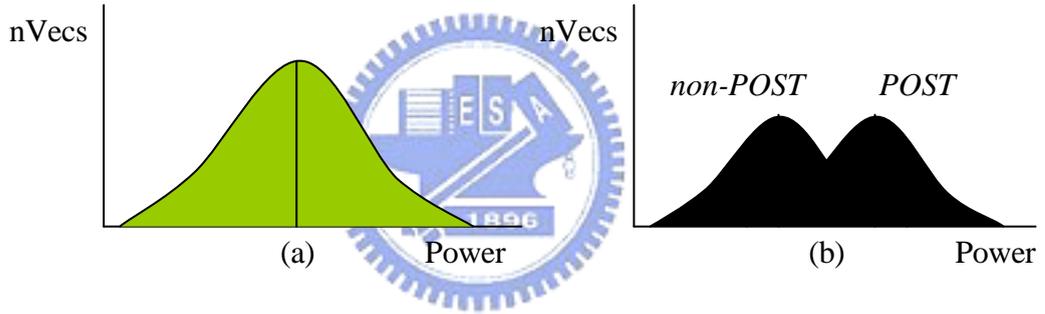


Figure 15: Stratification with POST

By assuming that the transition behavior and the power sensitivities of the other inputs are independent of In1, $Power(\mathbf{V}_1)$ and $Power(\mathbf{V}_2)$ are redefined as:

$$\hat{Power}(\mathbf{V}_{non-POST}) = A_{In1,non-POST} + k(\mathbf{V}_1), \quad (117)$$

$$\hat{Power}(\mathbf{V}_{POST}) = A_{In1,POST} + k(\mathbf{V}_2), \quad (118)$$

where

$$k(\mathbf{V}) = \mathbf{X}'(\mathbf{V})\mathbf{A} + b, \quad (119)$$

and

$$\mathbf{X}'(\mathbf{V}) = \begin{cases} [T_0(i, \mathbf{V}), T_1(i, \mathbf{V}), T_2(i, \mathbf{V})] & 1 \leq i \leq n_{in}, i \neq In1 \\ [0, 0, 0] & i = In1 \end{cases}. \quad (120)$$

Under the assumption that the transition behavior and the power sensitivities of the other input are independent of In1, the $k(\mathbf{V}_1)$ and $k(\mathbf{V}_2)$ are equal and will be designated as k for short. Substituting equation (117) and (118) into (103), (104), (105) and (106) we get:

$$\hat{\mathbf{m}}_1 = E[A_{In1,non-POST} + k], \quad (121)$$

$$\hat{\mathbf{s}}_1^2 = E\left[(A_{In1,non-POST} + k)^2\right] - \hat{\mathbf{m}}_1^2, \quad (122)$$

$$\hat{\mathbf{m}}_2 = E[A_{In1,POST} + k], \quad (123)$$

and

$$\hat{\mathbf{s}}_2^2 = E\left[(A_{In1,POST} + k)^2\right] - \hat{\mathbf{m}}_2^2. \quad (124)$$

Since the power sensitivities of In1 and In2 are measured constants, the above equations can be simplified with some deductions and become:

$$\hat{\mathbf{m}}_1 = A_{In1,non-POST} + E[k], \quad (125)$$

$$\hat{\mathbf{s}}_1^2 = E[k^2] - E[k]^2, \quad (126)$$

$$\hat{\mathbf{m}}_2 = A_{In1,POST} + E[k], \quad (127)$$

and

$$\hat{\mathbf{s}}_2^2 = E[k^2] - E[k]^2. \quad (128)$$

From equation (126) and (128), $\hat{\mathbf{s}}_1^2$ can be found the same as $\hat{\mathbf{s}}_2^2$. Therefore, they are designated as \mathbf{s}_k^2 because they are equal to the variance of k as well. Replacing the $\hat{\mathbf{s}}_1^2$ and $\hat{\mathbf{s}}_2^2$ in equation (102) with \mathbf{s}_k^2 ,

$$\left(\frac{\mathbf{s}_k}{\mathbf{m}_1}\right)^2 + \left(\frac{\mathbf{s}_k}{\mathbf{m}_2}\right)^2 < \frac{(n_1 + n_2)^2 \mathbf{s}_k^2}{(n_1 \mathbf{m}_1 + n_2 \mathbf{m}_2)^2} + \frac{(n_1 \mathbf{m}_1 - n_2 \mathbf{m}_2)^2}{(n_1 \mathbf{m}_1 + n_2 \mathbf{m}_2)^2}. \quad (129)$$

Since $\mathbf{m}_2 > \mathbf{m}_1$, equation (129) can be tightened by substituting the \mathbf{m}_1 of denominator of the first term at right hand side with \mathbf{m}_2 , then

$$\left(\frac{\mathbf{s}_k}{\mathbf{m}_1}\right)^2 + \left(\frac{\mathbf{s}_k}{\mathbf{m}_2}\right)^2 < \left(\frac{\mathbf{s}_k}{\mathbf{m}_2}\right)^2 + \frac{(n_1 \mathbf{m}_1 - n_2 \mathbf{m}_2)^2}{(n_1 \mathbf{m}_1 + n_2 \mathbf{m}_2)^2}. \quad (130)$$

After some deductions and taking the square root of both sides, the following relationship is obtained,

$$\mathbf{s}_k < \left| \frac{\mathbf{m}_1 (n_1 \mathbf{m}_1 - n_2 \mathbf{m}_2)}{(n_1 \mathbf{m}_1 + n_2 \mathbf{m}_2)} \right|. \quad (131)$$

By moving the \mathbf{s}_k to the right, the gain of separating the population according to the POST of an input is defined as,

$$Gain = \left| \frac{\mathbf{m}_1 (n_1 \mathbf{m}_1 - n_2 \mathbf{m}_2)}{(n_1 \mathbf{m}_1 + n_2 \mathbf{m}_2)} \right| - \mathbf{s}_k > 0. \quad (132)$$

This is the one and only equation for checking two populations should be separated or not. The rule of thumb observed from equation (132) is that, the more distant that \mathbf{m}_2 away from \mathbf{m}_1

the larger the gain is, and vice versa. With this rule, a heuristic method of stratifying the input vectors with POST is proposed.

Given the power sensitivities and the input statistics of a circuit, the input with the largest positive gain is first selected for dividing the population into two sub-populations. In each sub-population, the next input that has the largest positive gain is then selected for dividing the sub-population. The procedure is repeated until no input with positive gain is available.

4.5 Integration with SPICE3

SPICE is a *Simulation Program with Integrate Circuit Emphasis*. It solves the combination of KVL and KCL equations for the voltages of all nodes and the currents for some certain branches. The loading functions that translate each device into the KVL and KCL equations are listed in appendix B. In this section, the flow for the proposed power profiler will be detailed. Before that, the flow of the transient analysis that the proposed power profiler mainly based on will be demonstrated as following.

4.5.1 Transient Analysis Flow

To perform transient analysis, the circuit is analyzed at some time points selected between the starting point and the stopping point. The basic flowchart of the transient analysis is shown in Figure 17:

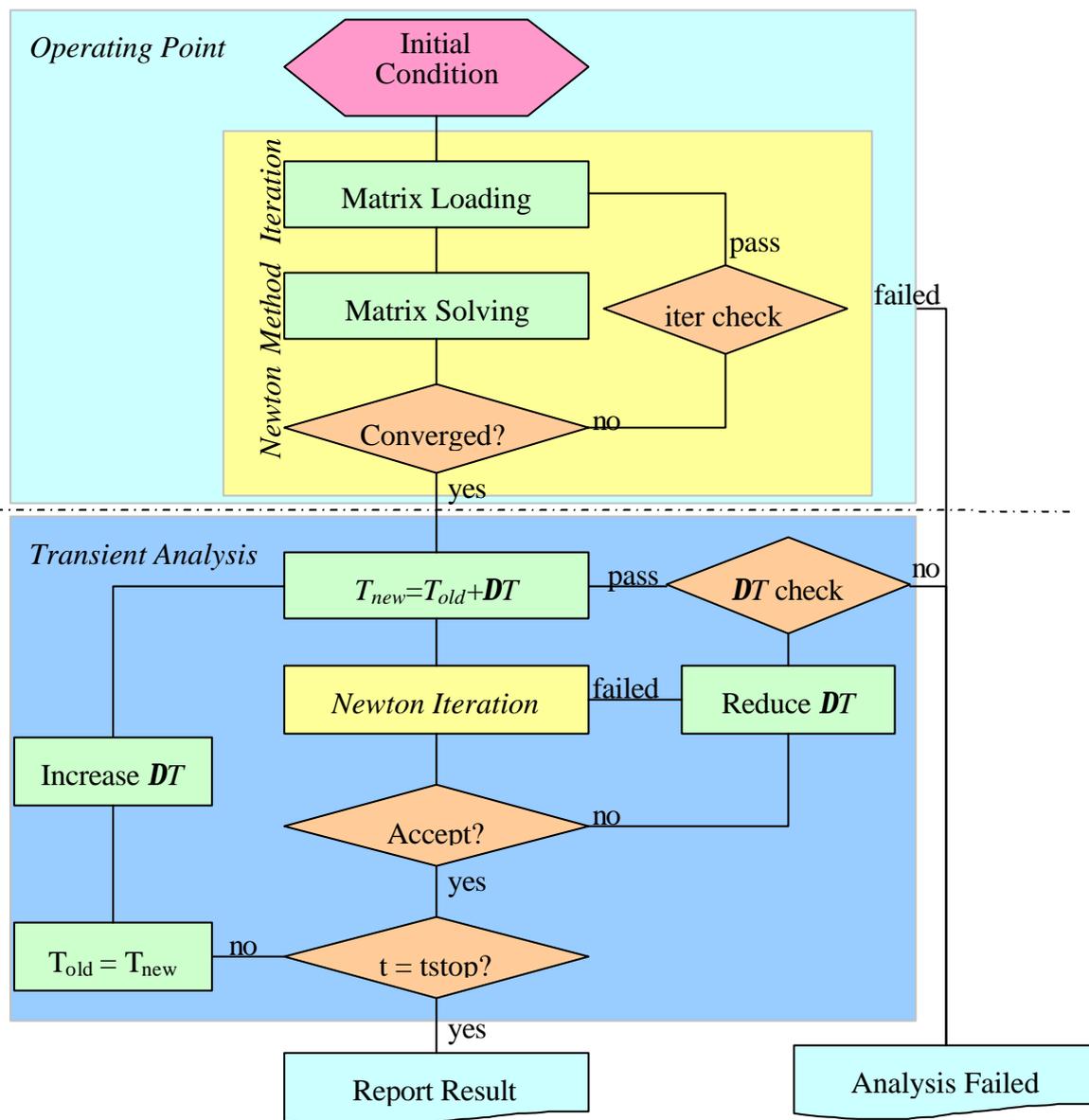


Figure 16: Flowchart of transient analysis in SPICE

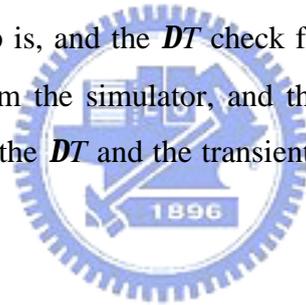
Operating Point Analysis

For the first time point, the initial conditions as well as the elements of the devices are loaded into the matrix. After that, the KCL and KVL equations are solved to get the nodal voltages and the branch currents of voltage sources and inductors. The ways of solving the matrix include *Gaussian Elimination* and *LU decompositions*. Both methods are with the same order of complexity. After matrix solving, the solutions are saved and another set of matrix loading and matrix solving are performed for checking the convergence of the solutions. If the solutions are within the tolerable range of the solutions of previous iteration, the new solutions are considered as converged. Otherwise, another iteration of matrix loading and

matrix solving is needed. This is called the *Newton's Iteration Method*. It has a limitation that the non-linear equations are monotonic functions around the initial guess. If the solutions do not converge to some values with a limited number of iterations, the *Newton Iteration Method* for operating point analysis is failed. In this circumstance, users have to check the initial conditions or to tune the simulation options for improving the DC convergence.

Transient Analysis

After the DC operating point is obtained, the transient analysis begins with a heuristic time step DT . The *Newton Iteration Method* is invoked to solve the new solutions of a new time point in the transient analysis at T_{new} . Every time the *Newton Iteration Method* fails, the time step is reduced to try to make the monotonic assumption stand. However, device models, especially the MOS models, are sometimes discontinuous if the model parameters are not properly designed and extracted. Under that circumstance, there may be no convergence no matter how small the time step is, and the DT check fails. An “internal time step too small” error message will pop up from the simulator, and the simulation halts. For most cases, a convergent result by shrinking the DT and the transient analysis continues until the stop time of the analysis is reached.



4.5.2 Automatic Power Profiler

The flow chart of the power profiling system is depicted in the following figure:

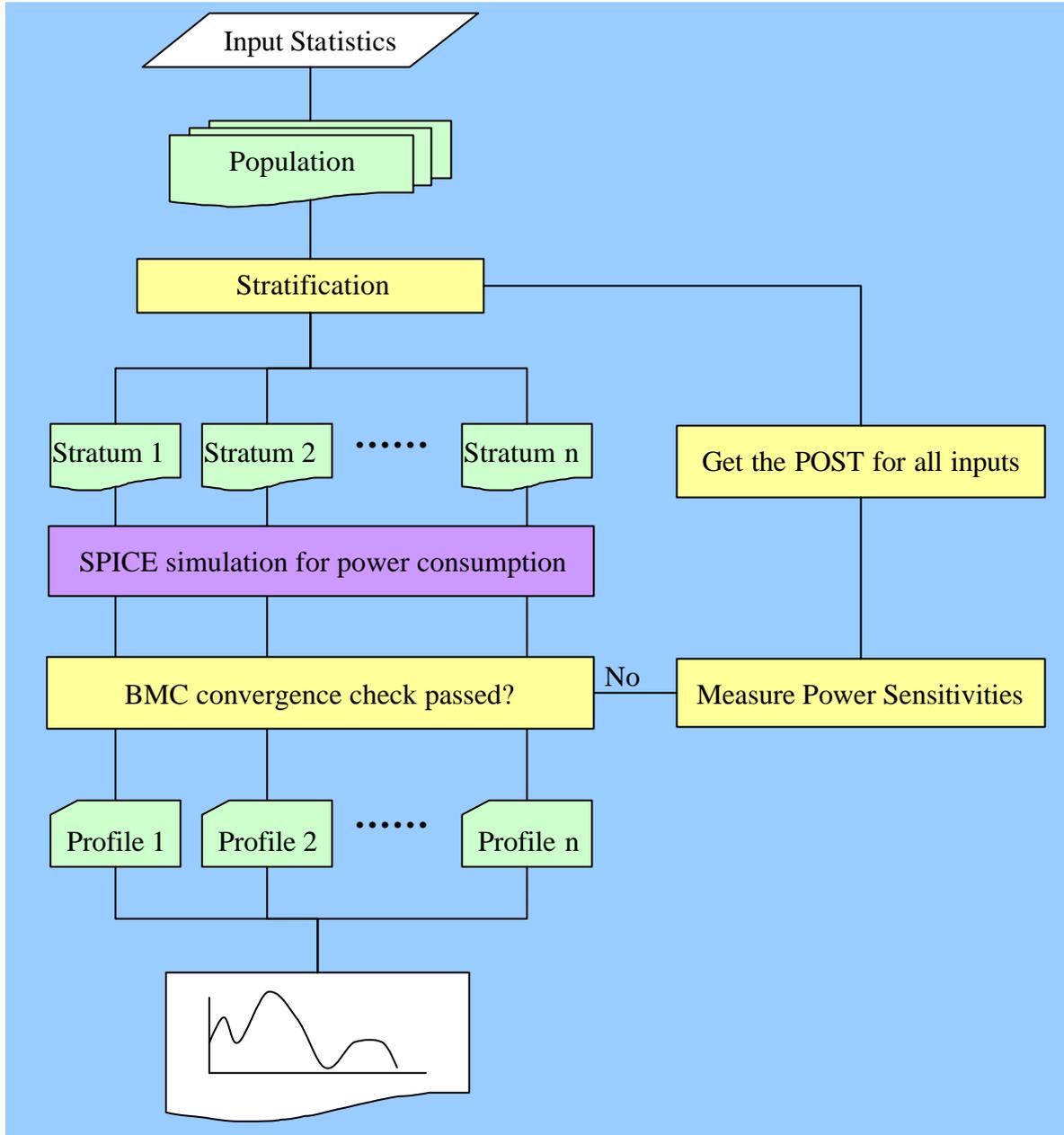


Figure 17: Flowchart of power profiler

4.6 Screen Shots

Here are some snapshots of running the tools and the graph that the tool will report. The power profiler takes normal SPICE netlist with some additional information including

1. A flag on the input voltage source that indicates it is a primary input.
2. The signal probability and the transition density of each primary input.
3. Option VDD for the definition of logic-1, (logic-0 is default grounded).
4. Option CLK for the definition of the period of a clock cycle.

5. Option SLEW for the definition of the slew rate of the signal transitions.
6. Option MONTE for turning on and off the Monte Carlo convergence criterion.
7. Option BOOTSTRAP for turning on and off the Bootstrap convergence criterion.
8. Option STRAT for turning on and off the stratification flow.

After setting up the netlist with the additional information, a simple command that evokes the power profiler as:

```
PowerPro    SPICE_NETLIST_FILE
```

And the messages on the screen are depicted in Figure 18:

```

Shell - Konsole
Session Edit View Settings Help
leon[13:30]% powerpro ckt

*****
***      PowerPro - written by Heng-Liang Huang      ***
***      Build date - 0405.30 09:23                 ***
*****

*** Hostname: leon ( Linux 2.4.18-3 )
*** Memory: 128M. Load: 0.07, 0.02, 0.00

*** Input file: ckt.sp

Operating Point Analysis:
iter: 16

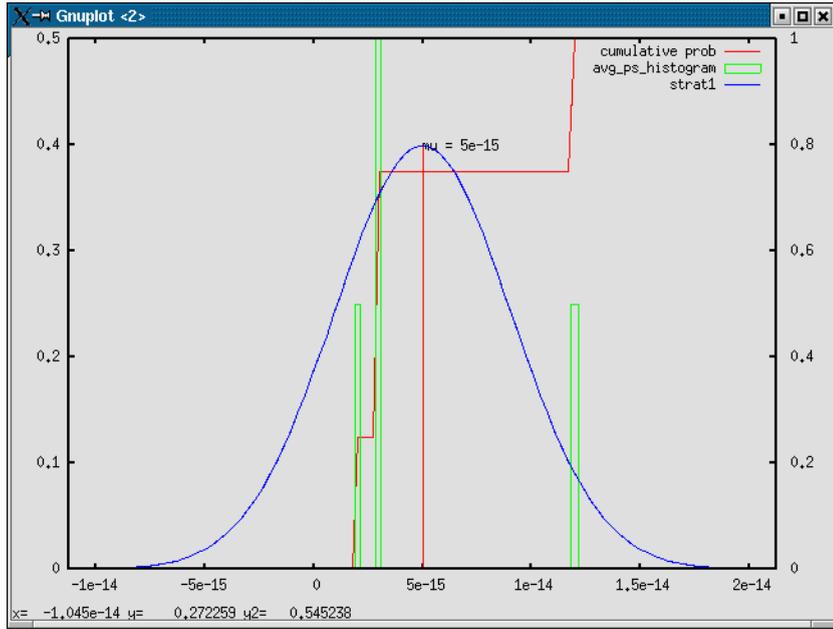
Transient Analysis:
* PowerPro *: Analysing      298.94ns of 310.00  ns
Restratified!
* PowerPro *: Analysing      320.00ns of 320.00ns
* PowerPro *: job done.

leon[13:31]%

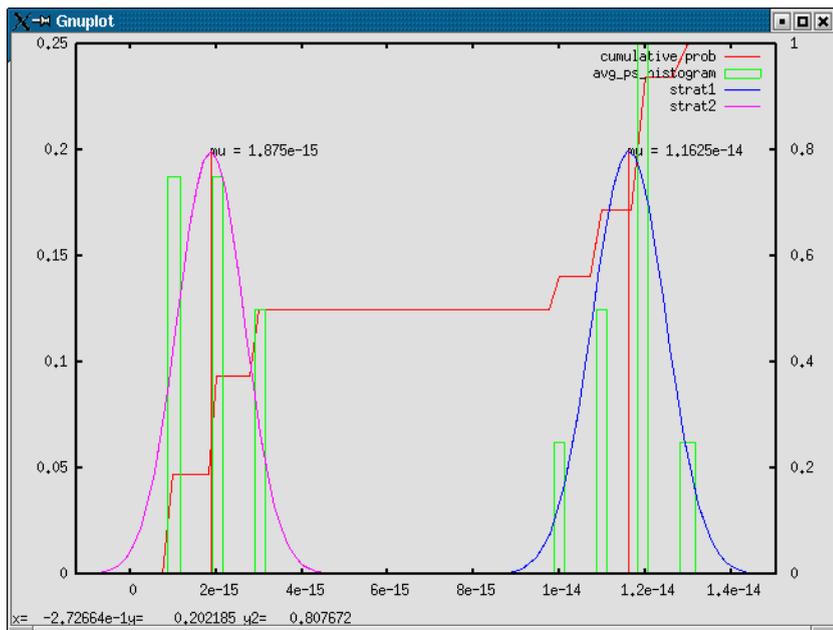
```

Figure 18: Screen log of power profiler

There is also a pop up window that plot the profile of the power consumption distribution shown in Figure 19.



(a)



(b)

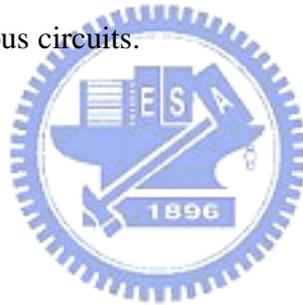
Figure 19: Power profile from power profiler

In Figure 19 (a), the stratification flow is not turned on. Therefore, the profile obtained from the profiler contains only one stratum. However, the power consumption distribution looks more like a bimodal distribution as in Figure 19 (b). In Figure 19 (b), the proposed automatic stratification flow has successfully stratified the population into two strata, and captured the bimodal nature of the distribution by showing the distribution with two normal distributions separately.

4.7 Summary

In this chapter, a novel definition POST was proposed for defining the input transitions that tend to induce higher power consumption. Furthermore, the POST is utilized for stratifying the input sequence with unlimited length so that the average power within each stratum can be accurately and efficiently estimated. This new stratification method with POST was also implemented in the most accurate simulator SPICE. A fully automated flow for calculating the profile of the power consumption distribution was also implemented.

Although we have put a lot of effort in implementation, there are still some improvements that can be done to make it more complete and solid. The power profiler at current stage targets combinational circuit only because sequential circuits are not suitable for vector sampling due to the high temporal correlation between the input vectors. Besides, PowerPro is implemented for synchronous circuits because there is no clear definition of the power corresponding to one input vector for asynchronous circuits.



5 Conclusion and Future Work

In this dissertation, the definition of power sensitivities was examined and proposed a method of selecting the nominal points for measuring the power sensitivities to increase the accuracy was proposed. With the knowledge of the meaning of power sensitivities, it is was utilized for the stratification of input vector sampling for simulation based power estimation. The Bootstrap Monte Carlo approached was proposed to improve the reliability of Monte Carlo simulation by calculating the confidence interval more accurately with the Bootstrap resampling technique. Since the power sensitivities measurement can be updated dynamically when simulating the sampled input vectors, the stratification process can be performed repeatedly whenever the power sensitivities are updated. Stratification with power sensitivities sum of each input vector is good for reducing the sample variances, although the calculating and sorting of the power sensitivities sums of the input vectors may be too complex when the input sequence is very long or even infinite. Therefore, a novel definition POST was proposed to define the transitions of inputs that tend to induce higher power consumption of the circuit. A heuristic stratification method based on POST was also proposed to reduce the number of sampled vectors required so that the power consumption distribution can be accurately and efficiently plotted. Both stratification and power estimation flow in this dissertation were implemented in the most accurate SPICE3 simulation engine from Berkeley. Users can get a visual image of the distribution of the power consumptions by specifying the primary input pins and their signal probabilities and transition densities. The theories and algorithms that are used or proposed in this dissertation are all demonstrated in the most practical and useful way through the implementation.

To sum up, power estimation is rather a measurement for further analysis than a number for meeting the specification. Here are some extensions that can be done on the platform of this dissertation.

Worst Case Sequence Generation

The worst case power consumption can be determined by finding one and only one input vector which induces the largest power consumption. However, defining the worst case power consumption with only one input vector could be too pessimistic and lost the generality. After stratifying the population with POST, the sub-population that expected to have the largest power consumption can be easily identified. It can also be used for the identification of the worst case operating mode of the circuit. A worst case input sequence will be able to be generated from the sub-population as well.

Reliability Analysis

Electromigration rules limit the average current densities that are allowed to flow in the metal segments in different layers. With the estimated power consumption, designers can check the electromigration rules for each metal segment. By changing the measurement scheme from power dissipation to sub-circuit port currents, the full chip IR drop analysis can also be implemented on the platform proposed in this dissertation.

Full Chip Temperature Profiling

With the estimated power dissipation of the sub-circuits, the temperature profile can be obtained by analyzing the positions of the heat sources and the efficiencies of the heat conduction paths. The spot that has potential heat problem can be identified for designer to rearrange the placement around hot spot.

6 References

- [1] L. Benini, M. Favalli and B. Ricco, "Analysis of Hazard Contributions to Power Dissipation in CMOS ICs", in proceedings of International Workshop on Lower Power Design, 1994, pp.27-32.
- [2] J. R. Black, "Electromigration-A Brief Survey and Some Recent Results", IEEE Transaction on Electron Devices, 1969, pp.338-347.
- [3] H. Chiueh, J. Draper, L. Luh and J. Choma Jr., "A Novel Model for On-chip Heat Dissipation", in proceedings of IEEE Asia-Pacific Conference on Circuit and Systems, 1998, pp.779-782.
- [4] Y. K. Cheng and S. M. Kang, "An Efficient Method for Hot-spot Identification in ULSI Circuits", in proceedings of International Conference on Computer Aided Design, 1999, pp.124-127.
- [5] A. H. Ajami, M. Pedram, and K. Banerjee, "Effects of Non-uniform Substrate Temperature on the Clock Signal Integrity", in proceedings of IEEE Custom Integrated Circuits Conference, 2001, pp.233-236.
- [6] A. H. Ajami, K. Banerjee and M. Pedram, "Non-uniform Chip-Temperature Dependent Signal Integrity", in proceedings of VLSI Technology Symposium, 2001, pp.145-146.
- [7] A. H. Ajami, K. Banerjee, M. Pedram, and L. P.P.P. van Ginneken, "Analysis of Non-Uniform Temperature-Dependent Interconnect Performance in High Performance ICs", in proceedings of ACM/IEEE Design Automation Conference, 2001, pp.567-572.
- [8] A. H. Ajami, K. Banerjee and M. Pedram, "Analysis of Substrate Thermal Gradient Effects on Optimal Buffer Insertion," in proceedings of International Conference on Computer Aided Design, 2001, pp.44-48.
- [9] A. H. Ajami, M. Pedram, and K. Banerjee, "Analysis and Optimization of Thermal Issues in High Performance VLSI", in proceedings of International Symposium on Physical Design, 2001, pp.230-237.
- [10] S. Vemuru, N. Scheinberg, and E. Smith, "Short-Circuit Power Dissipation Formulae for CMOS Gates", in proceedings of IEEE International Symposium on Circuits and Systems, 1993, pp.1333-1336.
- [11] TSMC, "Copper Interconnect Technology Performance and Reliability" brochure.
- [12] H.J.M. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits", IEEE Journal of Solid-State Circuits, Aug. 1984, pp.468-473.

- [13] S. Wang, "Fundamentals of Semiconductor Theory and Devices Physics", Prentice Hall, 1989, Chapter 1-6.
- [14] Hyper physics, "<http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html>".
- [15] Microwaves101.com, "Conductors and Resistors Materials", Feb 2004, in <http://www.microwaves101.com/encyclopedia/conducting.cfm>
- [16] E. S. Yang, "Microelectronic Devices", McGraw Hill, 1988, appendix A.
- [17] N. NS, F. Cano, H. Haznedar and D. Young, "A Practical Approach to Static Signal Electromigration Analysis", in proceedings of ACM/IEEE Design Automation Conference, 1998, pp.572-577.
- [18] M. K. Anderson, "A Normality Test for the Mean Estimator", Working Paper Series in Economics and Statistics No. 310, March 4 1999.
- [19] S. R. Huang, "Effectiveness of Optimum Stratified Sampling and Estimation in Monte Carlo Production Simulation", IEEE Transaction on Power System, May 1997, pp.566-572.
- [20] I. Miller, J. E. Freund, and R. A. Johnson, "Probability and statistics for engineers", Prentice Hall, 1990.
- [21] H. Stark and J. W. Woods, "Probability, Random Process, and Estimation Theory for Engineers", Prentice-Hall 1986.
- [22] L. Benini and G. D. Micheli, "System-Level Power Optimization: Techniques and Tools", in proceedings of International Symposium on Low Power Electronics and Design, 1999, pp.288-293.
- [23] O. Coudert, R. Haddad and K. Keutzer, "What is the State of the Art in Commercial EDA Tools for Low Power?", in proceedings of International Symposium on Low Power Electronics and Design, 1996, pp.181-187.
- [24] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits", IEEE Transaction VLSI, Dec. 1994, pp.446-455.
- [25] P. E. Landman, J. M. Rabaey, "Power Estimation for High Level Synthesis", IEEE European Conference on Design Automation 1993, pp.361-366.
- [26] P. Landman, "High-Level Power Estimation", in proceedings of International Symposium on Low Power Electronics and Design, 1996, pp.29-35.
- [27] F.N. Najm, "Towards a High-Level Power Estimation Capability", in proceedings of International Symposium on Low Power Electronics and Design, 1995, pp.87-92.

- [28] D. Marculescu, R. Marculescu, and M. Pedram, "Information Theoretic Measures for Power Analysis", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Jun. 1996, pp.599-610.
- [29] E. Macii, M. Pedram and F. Somenzi, "High-Level Power Modeling Estimation, and Optimization", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems Nov. 1998, pp.1061-1079.
- [30] F. Ferrandi, F. Fummi, E. Macii, and M. Poncino, "Power Estimation of Behavioral Descriptions" in proceedings of Design, Automation and Test in Europe, 1998, pp.762-766.
- [31] S. Devadas, K. Keutzer and J. White, "Estimation of Power Dissipation in CMOS Combinational Circuits", in proceedings of IEEE Custom Integrated Circuits Conference, 1990, pp.19.7.1-19.7.6.
- [32] A. Shen, A. Ghosh, S. Devadas, K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks", in proceedings of International Conference on Computer Aided Design, 1992, pp.402-407.
- [33] S. Devadas, K. Keutzer and J. White, "Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Mar. 1992, pp.373-383.
- [34] A. Ghosh, S. Devadas, K. Keutzer and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits", in proceedings of ACM/IEEE Design Automation Conference, 1992, pp.253-259.
- [35] J. Monterio, S. Devadas, B. Lin, C.Y. Tsui, and M. Pedram, "Exact and Approximate Methods of Switching Activity Estimation in Sequential Logic Circuits", in proceedings of International Workshop on Lower Power Design, 1994, pp.117-122.
- [36] J. Monteiro, S. Devadas and B. Lin, "A Methodology for Efficient Estimation of Switching Activity in Sequential Logic Circuits", in proceedings of ACM/IEEE Design Automation Conference, 1994, pp.12-17.
- [37] A. Shen, S. Devadas, A. Ghosh, "Probabilistic Manipulation of Boolean Functions Using Free Boolean Diagrams", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Jan. 1995, pp.87-95.
- [38] T. Uchino, F. Minami, T. Mitsuhashi and N. Goto, "Switching Activity Analysis using Boolean Approximation Method", in proceedings of International Conference on Computer Aided Design, 1995, pp.20-25.

- [39] G. I. Stamoulis, I. N. Hajj, "Improved Techniques for Probabilistic Simulation Including Signal Correlation Effects", in proceedings of ACM/IEEE Design Automation Conference, 1993, pp.379-383.
- [40] H. Choi, J. H. Yi and S. H. Hwang, "Estimation of Inertial Delay Dependent Switching Activities by Using Time-Stamped Transition Density", in proceedings of IEEE International Symposium on Circuits and Systems, 1997, pp.1528-1531.
- [41] H. L. Huang, J.Y. Lin, W. Z. Shen and J. Y. Jou, "A New Method for Constructing IP Level Power Model Based on Power Sensitivity", in proceedings of Asia and South Pacific Design Automation Conference, 2000, pp.135-140.
- [42] S.Y. Yuan, K.H. Chen, J.Y. Jou and S.Y. Kuo, "Static Power Analysis for Power-Driven Synthesis", IEE proceeding of Computers and Digital Techniques, Mar 1998, pp 89-95.
- [43] G. Markowsky, "Bounding Signal Probabilities in Combinational Circuits", IEEE Trans on Computers, Oct. 1987, pp.1247-1251.
- [44] F. N. Najm, "Transition Density: A New Measure of Activity in Digital Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Feb. 1993, pp.310-323.
- [45] F. N. Najm, "Improved Estimation of the Switching Activity for Reliability Prediction in VLSI Circuits", in proceedings of IEEE Custom Integrated Circuits Conference 1994, pp.429-432.
- [46] F. N. Najm, "Low-Pass Filter for Computing the Transition Density in Digital Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Sep. 1994, pp.1123-1131.
- [47] S. Gupta and F. N. Najm, "Analytical Model for High Level Power Modeling of Combinational and Sequential Circuits", in proceedings of Alessandro Volta Memorial Workshop on Low Power Design, 1999, pp164-172.
- [48] H. Mehta, M. Borah, R. M. Owens, M. J. Irwin, "Accurate Estimate of Combinational Circuit Activity", in proceedings of ACM/IEEE Design Automation Conference 1995, pp.618-622.
- [49] A. Papoulis, "Probability and Statistics", Prentic-Hall, 1990.
- [50] R. Marculescu, D. Marculescu and M. Pedram, "Switching Activity Analysis Considering Spatiotemporal Correlations", in proceedings of International Conference on Computer Aided Design, 1994, pp.294-299.

- [51] D. Marculescu, R. Marculescu and M. Pedram, "Sequence Compaction for Probabilistic Analysis of Finite-State Machine", in proceedings of ACM/IEEE Design Automation Conference, 1997, pp.12-15.
- [52] C.S. Ding, C.Y. Tsui and M. Pedram, "Gate-Level Power Estimation Using Tagged Probabilistic Simulation", IEEE Transaction CAD, Nov. 1998, pp.1099-1107.
- [53] D. Marculescu, R. Marculescu and M. Pedram, "Theoretical Bounds for Switching Activity Analysis in Finite-State Machines", in proceedings of International Symposium on Low Power Electronics and Design, 1998, pp.36-41.
- [54] S. Ercolani, M. Favalli, M. Damiani, P. Olivo and B. Ricco, "Estimate of Signal Probability in Combinational Logic Networks", in proceedings of European Test Conference, 1989, pp.132-138.
- [55] T.L. Chou, K. Roy, S. Prasad, "Estimation of Circuit Activity Considering Signal Correlations and Simultaneous Switching", in proceedings of International Conference on Computer Aided Design, 1994, pp.300-303.
- [56] T.L. Chou and K. Roy, "Statistical Estimation of Sequential Circuit Activity", in proceedings of International Conference on Computer Aided Design, 1995, pp.34-37.
- [57] Y.J. Lim, K.I. Son, H.J. Park, and M Soma, "A Statistical Approach to the Estimation of Delay Dependent Switching Activities in CMOS Combinational Circuits", in proceedings of ACM/IEEE Design Automation Conference, 1996, pp.445-450.
- [58] D. E. Wallace, "High-Level Delay Estimation for Technology-Independent Logic Equations", in proceedings of International Conference on Computer Aided Design, 1990, pp.188-191.
- [59] J.M. Jou, S.C. Chen and C.L. Wang, "Fast Delay Dependent Power Estimation of Large Combinational Circuits", in proceedings of IEEE International Symposium on Circuits and Systems 1998, pp.53-56.
- [60] J. H. Satyanarayana and K. K. Parhi, "HEAT: Hierarchical Energy Analysis Tool", in proceedings of ACM/IEEE Design Automation Conference, 1996, pp.9-14.
- [61] A. Bogliolo, L. Benini, G. D. Micheli, "Characterization-Free Behavioral Power Modeling", in proceedings of Design, Automation and Test in Europe, 1998, pp.767-773.
- [62] M. Nemani and F. N. Najm, "Towards a High-Level Power Estimation Capability", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 1996, pp.588-598.
- [63] S. Gupta and F. N. Najm, "Power Macromodeling for High Level Power Estimation", in proceedings of ACM/IEEE Design Automation Conference 1997, pp.365-370.

- [64] R. Marculescu, D. Marculescu, and M. Pedram, "Probabilistic Modeling of Dependencies During Switching Activity Analysis", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Feb. 1998, pp.73-83.
- [65] C. Y. Tsui, M. Pedram and A. M. Despain, "Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs", in proceedings of ACM/IEEE Design Automation Conference, 1994, pp.18-23.
- [66] D. I. Cheng, M. M. Sadowska, K. T. Cheng, "Speeding Up Power Estimation by Topological Analysis", in proceedings of IEEE Custom Integrated Circuits Conference, 1995, pp.623-626.
- [67] J. C. Costa, J. C. Monterio and S. Devadas, "Switching Activity Estimation using Limited Depth Reconvergent Path Analysis", in proceedings of International Symposium on Low Power Electronics and Design, 1997, pp.184-189.
- [68] R. Radjassamy and J. D. Carothers, "Vector Compaction for Efficient Simulation-Based Power Estimation", in proceedings of IC/Package Design Integration 1998, pp.138-142.
- [69] C.Y. Tsui, M. Pedram and A. M. Despain, "Efficient Estimation of Dynamic Power Consumption under a Real Delay Model", in proceedings of International Conference on Computer Aided Design, 1993, pp.224-228.
- [70] C.S.Ding and M. Pedram, "Tagged Probabilistic Simulation Provides Accurate and Efficient Power Estimation at Gate Level", in proceedings of IEEE Symposium of Low Power Electronics, 1995, pp.42-43.
- [71] Z. Chen and K. Roy, "Sensitivity of Power Dissipation to Uncertainties in Primary Input Specification", in proceedings of IEEE Custom Integrated Circuits Conference, 1997, pp.23.4.1-23.4.4.
- [72] Z. Chen, K. Roy and T.L. Chou, "Sensitivity of Power Dissipation to Uncertainties in Primary Input Specification", in proceedings IEEE Custom Integrated Circuits Conference 1997, pp.487-490.
- [73] Z. Chen, K. Roy, and T. L. Chou, "Power Sensitivity – A New Method to Estimate Power Dissipation Considering Uncertain Specifications of Primary Inputs", in proceedings International Conference on Computer Aided Design, 1997, pp.40-44.
- [74] Z. Chen, K. Roy and T.L. Chou, "Efficient Statistical Approach to Estimate Power Considering Uncertain Properties of Primary Inputs", IEEE Transaction on VLSI System, Sep. 1998, pp.484-492.

- [75] Z. Chen, K. Roy, and K.P. Chong, "Estimation of Power Sensitivity in Sequential Circuits with Power Macromodeling Application", in proceedings of International Conference on Computer Aided Design, 1998, pp.468-472.
- [76] C.Y. Wang and K. Roy, "Maximum Power Estimation for CMOS Circuits Using Deterministic and Statistical Approaches", IEEE Transaction on VLSI System, Mar. 1998, pp.134-140.
- [77] Z. Chen and K. Roy, "A Power Macromodeling Technique Based on Power Sensitivity", in proceedings ACM/IEEE Design Automation Conference, 1998, pp.678-683.
- [78] C.Y. Tsui, R. Marculescu, D. Marculescu, M. Pedram, "Improving the Efficiency of Power Simulators by Input Vector Compaction", in proceedings of ACM/IEEE Design Automation Conference, 1996, pp.165-168.
- [79] A. Pinar, and C. L. Liu, "Power Invariant Vector Sequence Compaction", in proceedings of International Conference on Computer Aided Design, 1998, pp473-476.
- [80] N. Dragone, R. Zafalon, C. Guardiani and C. Silvano, "Power Invariant Vector Compaction Based on Bit Clustering and Temporal Partitioning", in proceedings of International Symposium on Low Power Electronics and Design, 1998, pp118-120.
- [81] S.C. Fang, and S. Puthenpura, "Linear Optimization and Extensions, Theory and Algorithms", Prentice Hall, 1993.
- [82] R. L. Rardin, "Optimization in Operations Research", Prentice Hall, 1998.
- [83] A. Bogliolo, L. Benini and G.D. Micheli, "Adaptive Least Mean Square Behavioral Power Modeling", in proceedings European Design and Test Conference, pp.404-410, 1997.
- [84] A. Bogliolo and L. Benini, "Node Sampling: A Robust RTL Power Modeling Approach", in proceedings International Conference on Computer Aided Design, pp.461-467, 1998.
- [85] R. Radjassamy and J.D. Carothers, "Vector Compaction for Efficient Simulation-Based Power Estimation", in IEEE Symposium on IC/Package Design Integration 1998, pp.138-142.
- [86] S. Y. Huang, K. C. Chen, K. T. Cheng, and T. C. Lee, "Compact Vector Generation for Accurate Power Simulation", in proceedings ACM/IEEE Design Automation Conference 1996, pp.161-164.
- [87] B. Efron, "Better Bootstrap Confidence Intervals", Journal of the American Statistical Association 1987, pp.171-200.
- [88] B. Efron, and R. J. Tibshirani "An Introduction to the Bootstrap", Chapman & Hall, 1993.

- [89] A. Macii, E. Macii, M. Poncino and R. Scarsi, "Stream Synthesis for Efficient Power Simulation Based on Spectral Transforms", in proceedings of IEEE Symposium on Low Power Electronics and Design 1998, pp.30-35.
- [90] Z. Chen, M. Johnson, L. Wei and K. Roy, "Estimation of Standby Leakage Power in CMOS Circuits Considering Accurate Modeling of Transistor Stacks", in proceedings of International Symposium on Low Power Electronics and Design, 1998, pp.239-244.
- [91] A. Bogiolo, L. Benini, B. Ricco, "Power Estimation of Cell-Based CMOS Circuits", in proceedings of ACM/IEEE Design Automation Conference, 1996, pp.433-438.
- [92] L. Benini, G. D. Micheli, E. Macii, M. Poncino and R. Scarsi, "Fast Power Estimation for Deterministic Input Streams", in proceedings of International Conference on Computer Aided Design, 1997, pp.494-501.
- [93] Y.M. Jiang, S.Y. Huang, K.T. Cheng, D.C. Wang, C.Y. Ho, "A Hybrid Power Model for RTL Power Estimation", in proceedings of Asia and South Pacific Design Automation Conference, 1998, pp.551-556.
- [94] R. P. Llopis and K. Goossens, "The Petrol Approach to High-Level Power Estimation", in proceedings of International Symposium on Low Power Electronics and Design, 1998, pp.130-132.
- [95] A. Bogliolo, L. Benini, and G. D. Micheli, "Adaptive Least Square Behavioral Power Modeling", in proceedings of European Design and Test Conference, 1997, pp.400-406.
- [96] L. P. Yuan, C. C. Teng, and S. M. Kang, "Statistical Estimation of Average Power Dissipation in CMOS VLSI Circuits Using Nonparametric Techniques", in proceedings of International Symposium on Low Power Electronics and Design, 1996, pp.73-78.
- [97] R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A Monte Carlo Approach for Power Estimation", IEEE Transaction on VLSI System, March 1993, pp.63-71.
- [98] F. N. Najm, "Simulation and Modeling – Estimating Power Dissipation in VLSI Circuits", IEEE Circuit and Device Magazine, 1994, pp.11-19.
- [99] M. G. Xakellis and F. N. Najm, "Statistical Estimation of the Switching Activity in Digital Circuits", in proceedings of ACM/IEEE Design Automation Conference, 1994, pp.728-733.
- [100] V. Saxena, F. N. Najm, and I. N. Hajj, "Monte-Carlo Approach for Power Estimation in Sequential Circuits", in proceedings of European Design and Test Conference, 1997, pp.416-420.

- [101] M. M. Khellah and M. I. Elmasry, "Effective Capacitance Macro-Modeling for Architectural for Architectural-Level Power Estimation", in proceedings Great Lakes Symposium on VLSI, pp.414-419, 1998.
- [102] C. S. Ding, C. T. Hsieh, Q. Wu, and M. Pedram, "Stratified Random Sampling for Power Estimation", in proceedings of International Conference on Computer Aided Design, 1996, pp.576-582.
- [103] C. S. Ding, Q. Wu, C. T. Hsieh, and M. Pedram, "Stratified Random Sampling for Power Estimation", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, June 1998, pp.465-478.
- [104] R. Marculescu, D. Marculescu, and M. Pedram, "Efficient Power Estimation for Highly Correlated Input Streams", in proceedings of ACM/IEEE Design Automation Conference, 1995, pp.628-634.
- [105] R. Marculescu, D. Marculescu and M. Pedram, "Adaptive Models for Input Data Compaction for Power Simulators", in proceedings of Asia and South Pacific Design Automation Conference, 1997, pp.391-396.
- [106] R. Marculescu, D. Marculescu, and M. Pedram, "Hierarchical Sequence Compaction for Power Estimation", in proceedings ACM/IEEE Design Automation Conference 1997, pp.570-575.
- [107] C.S. Ding, C.T. Hsieh, and M. Pedram, "Improving Sampling Efficiency for System Level Power Estimation", in proceedings of International Symposium on Low Power Electronics and Design, 1998, pp.115-117.
- [108] R. Marculescu, D. Marculescu, and M. Pedram, "Sequence Compaction for Power Estimation Theory and Practice", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, July 1999, pp.973-993.
- [109] B. S. Amrutur, "Design and Analysis of Fast Low Power SRAMs", PHD dissertation of Stanford University, Aug. 1999.
- [110] M. Chinosi, R. Zafalon and C. Guardiani, "Automatic Characterization and Modeling of Power Consumption in Static RAMS", in proceedings of International Symposium on Low Power Electronics and Design, 1998, pp.112-114.
- [111] P. Hicks, M. Walnock, R. M. Owens, "Analysis of Power Consumption in Memory Hierarchies", in proceedings of International Symposium on Low Power Electronics and Design, 1997, pp.239-242.