Graphical Probabilistic Switching Model: Inference and Characterization for Power Dissipation in VLSI Circuits

by

Shiva Shankar Ramani

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Sanjukta Bhanja, Ph.D.
Yun-Leei Chiou, Ph.D.
Wilfrido A. Moreno, Ph.D.

Date of Approval:
September 8, 2004

Keywords: Probabilistic model, Simulation, Inference, Sampling

DEDICATION

In memory of my parents

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to my advisor Dr. Sanjukta Bhanja for her constant encouragement, support and guidance throughout the course of this research. She has helped me a lot to mould myself as a researcher. I would also like to express my deep appreciation to my committee members Dr. Yun-Leei Chiou and Dr. Wilfrido A. Moreno for their valuable time. My Sincere thanks to Dr. Yun-Leei Chiou for providing me with research projects during the initial part of my graduate studies.

I am grateful to my uncle for staying by me during the hard times and further assisting me financially. I would also like to take this opportunity to thank Essie Utley, Barbara Roberts, Jenevie Brown, Mina, and Eva for supporting and providing moral support throughout my stay with the UMSA. I am really grateful for the invaluable support and motivation that I received from my family and friends (especially my room-mates).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GRAPHICAL PROBABILISTIC SWITCHING MODEL: INFERENCE AND CHARACTERIZATION FOR POWER DISSIPATION IN VLSI CIRCUITS

Shiva Shankar Ramani

ABSTRACT

Power dissipation in a VLSI circuit poses a serious challenge in present and future VLSI design. A switching model for the data dependent behavior of the transistors is essential to model dynamic, load-dependent active power and also leakage power in active mode - the two components of power in a VLSI circuit. A probabilistic Bayesian network based switching model can explicitly model *all* spatio-temporal dependency relationships in a combinational circuit, resulting in zero-error estimates. However, the space-time requirements of exact estimation schemes, based on this model, increase with circuit complexity [5, 24]. This work explores a non-simulative, importance sampling based, probabilistic estimation strategy that scales well with circuit complexity. It has the any-time aspect of simulation *and* the input pattern independence of probabilistic models. Experimental results with ISCAS'85 benchmark shows order of magnitude reduction in maximum error, standard deviation specially for larger benchmarks with significantly low time. We also present a novel probabilistic method that is *not* dependent on the pre-specification of input-statistics or the availability of input-traces, to identify nodes that are likely to be leaky even in the active zone. This work emphasizes on stochastic data dependency and characterization of the input space, targeting data-dependent leakage power. The central theme of this work lies in obtaining the posterior input data distribution, *conditioned* on the leakage at an individual signal. We propose a minimal, causal, graphical probabilistic model (Bayesian Belief Network) for computing the posterior, based on probabilistic propagation flow against the causal direction, i.e. towards the input. We also provide two entropy-based measures to characterize the amount of uncertainties in the posterior input space as an indicator of the likelihood of the leakage of a signal. Results on ISCAS'85 benchmark shows that conclusive judgments can be made on many nodes without any prior knowledge about the input space.

CHAPTER 1

INTRODUCTION

1.1 Need for Low Power VLSI design

Gordon Moore (Moore's law) predicted an exponential growth in the number of transistors per square inch, nearly double the number each year and further expected this trend to continue for the foreseeable future [Figure 1.1.]. This drastic increase in chip density, together with decrease in feature sizes have made power dissipation a major issue in VLSI circuits. The increase in device density every year also demands for high operating frequency. As a result, the amount of power disspated per unit area or the power density is bound to increase which necessiates the use of costly packaging and heat sinks to keep the temperature levels of the chip within its limits. A stage has reached were we have to start analyzing our designs for power apart from area and speed constraints for better implementation. This has made Low Power Design the focus of VLSI research and development over the last decade. Another factor that drives the need for low power design is the rapidly increasing demand for portable electronic systems, which imposes severe restrictions on its size, weight and power. Battery life plays a major role when it comes to making a choice of a particular portable electronic item. The specific weight, which is the stored energy per unit weight, of a battery is not expected to have a revolutionary change that meets with the expanding applications of portable systems. Hence, estimation and optmization of a design for power apart from area and time as constraints becomes absolutely necessary to meet with the demands in portable systems design.

To summaraize, the need for low power design is due to following reasons:

- *Reduced battery life* due to high energy consumed by VLSI circuits.
- *Reduced reliability and speed* due to increase in power dissipation.
- *Increased costs* due to additional packaging and cooling system to reduce temperature.
- *Environmental concerns* due to unnecessary energy consumption and heat.

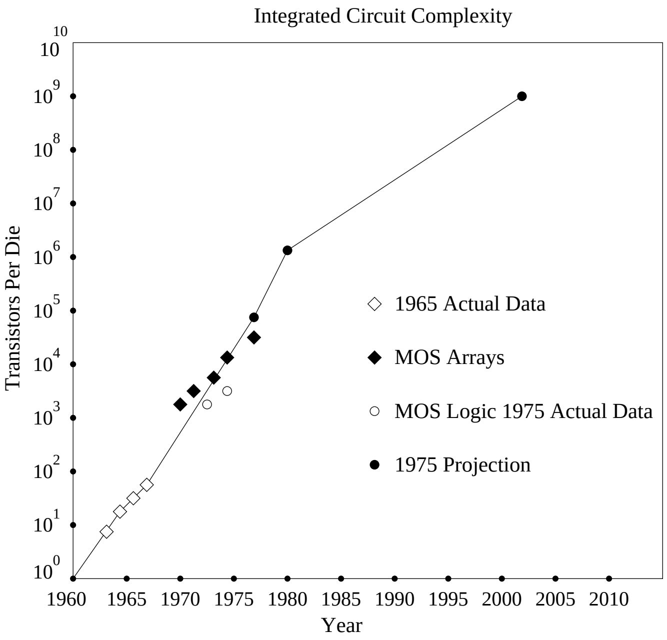


Figure 1.1. Graph depicting Moore's Law [1].

1.2 Components of Power dissipation

This section is devoted to give an overview on various sources of power dissipation in CMOS circuits. Average power consumption in CMOS circuits is due to 2 components. They are dynamic power and static (leakage) power. Average power can be expressed by the following equation

$$P_{avg} = P_{dynamic} + P_{leakage} \tag{1.1}$$

1.2.1 Dynamic power dissipation

Dynamic power consumption arises due to frequent charging and discharging of the parasitic capacitance during switching. So far, dynamic power has been the dominant component of power and accounts for about 75% of the total power dissipation. Interested reader is directed to [3] for detailed understanding on the derivations for dynamic power.

If the charging and the discharging cycle takes place at a frequency f, the total power dissipated in an inverter is given as

$$P_{dynamic} = E_s f = C_L V^2 f \tag{1.2}$$

The above equation Equation. 1.2 is an important equation in CMOS VLSI design representing the dynamic power dissipiation of a single gate with load capacitance C_L. In general, the total power of a circuit with n gates is given as:

$$P_{dynamic} = \sum_i C_i V_i^2 f_i \tag{1.3}$$

Voltage V_i in the Equation. 1.3 is same for all gates in a circuit and f_i is the frequency of switching for a particular gate. From the Equation 1.3, we can notice that the dynamic power is directly dependent on the frequency of switching. Hence an active circuit will dissipate more power than an idle circuit. Accurate estimation of dynamic power requires a careful analysis on the switching profile of each gate. First part of this thesis is focussed toward estimating the switching activity at gate level under zero delay assumption.

Another component of power that is caused due to input signal switching is the *short-circuit power* consumption. This component is due to a direct path from V_{dd} to ground, when both nmos and pmos conduct for a short while during switching. Consider an inverter circuit shown in the Figure 1.2.. When the input signal switches, a brief period exists during which both transistors, that is, nmos and pmos conduct. The reason is, PMOS turns on if the input signal level is below V_{tp} (PMOS threshold voltage) whereas for NMOS the input signal level has to be above V_{tn} (NMOS threshold voltage), as shown in Figure 1.2., thus causing a direct flow of current from the voltage source to ground. The power dissipated during the input signal transition phase is referred to as the short circuit power and is given by Equation. 1.4 [4]

$$P_{short} = K \frac{\beta}{12} (V_{dd} - 2V_T)^3 f \tau \tag{1.4}$$

β is the gain factor of a MOS transistor, V_T is the threshold voltage, and τ is the rise/fall time of the gate inputs. Short circuit power accounts for 10% of the overall power dissipation and is usually not taken into consideration during low power design. Factors that affect short circuit power are given below:

- The slope and duration of the input signal.
- The output loading capacitance.
- It also depends on process technology, temperature, etc.

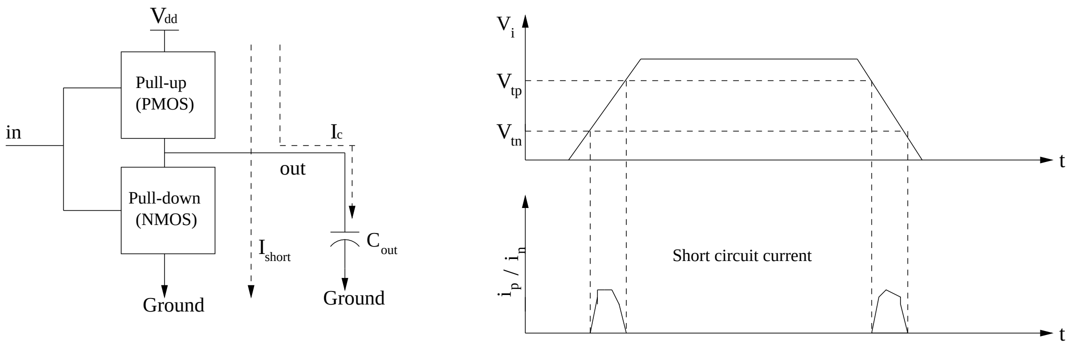


Figure 1.2. CMOS Inverter and a Graph showing its short circuit current [3].

Dynamic and Short circuit power dissipation depend on the switching acticity and hence in idle state (switching activity is zero) the circuit should actually not consume any power. In reality, there is another component called leakage power that causes power dissipation in the sleep or static mode.

1.2.2 Leakage power dissipation

Second component of power dissipation is the *Static leakage power consumption.* The reason for this type of dissipiation can be attributed to reverse bias diode leakage, sub-threshold leakage, gate oxide tunneling, leakage due to hot carrier injection, Gate-Induced Drain Leakage (GIDL), and channel punchthrough. Note that this type of power dissipation depends on the logic states of a circuit than its switching activites. Currently, power dissipated due to leakage is not significantly high but is expected to increase by year 2005 as the technology moves towards nano dimensions. The total leakage power dissipation is given by Equation 1.5.

$$P_{leakage} = (I_{diode} + I_{subthreshold} + I_{oxide-tunneling} + I_{hot-carrier} + I_{GIDL} + I_{channel-punchthrough}).V_{dd} \quad (1.5)$$

1. *Reverse Biased PN-junction* (I_1): The diode leakage is due to the formation of PN junctions between source or drain of the transistor and the bulk (substrate). Leakage current flows from the junction to the substrate when the diode is reverse biased. The magnitude of the current depends on process parameters, area of the PN junction, bias voltage, and temperature. Equation 1.6 gives diode leakage current.

$$I_{diode} = (I_s * (e^{V/V_{th}} - 1)) \quad (1.6)$$

- I_s is the reverse saturation current and it is dependent on temperature. I_s doubles for every ten degree increase in temperature.
- V_{th} is the thermal voltage, which is given by kT/q.

The reverse saturation current I_s is of the order of $1-5pA/\mu m^2$. Note that the diode leakage current occurs even during stand-by mode, that is, when there is no switching. Hence, the power disspation due to this mechanism will have a significant impact on a large chip containing several million transistors.

Note that for heavily doped p and n regions, the *BTBT* (band to band tunneling) dominates the diode leakage [15]. High field across the reverse biased pn junction causes a significant flow of electrons, by the tunneling process, from the valence band of the p-region to the conduction band of the n-region.

2. *Subthreshold Channel Leakage* (I_2): The primary contributor to leakage power is the sub-threshold or weak inversion conduction current. Strictly speaking, when a transistor is in off state, there should not be any current in the channel. But in reality, there is a non-zero current flowing through the channel as shown in Figure 1.3.. Hence the term sub-threshold leakage as it occurs at a voltage level well below the gate voltage. As the device dimension scales down, the power contributed by sub-threshold leakage becomes enormous and it exhibits an exponential dependence on the gate voltage. Subthreshold current is given by equation 1.7.

$$I_{subthreshold} = (I_0 * (e^{\frac{(V_{gs}-V_t)}{(\alpha V_{th})}}) \tag{1.7}$$

- V_t is the threshold voltage.
- I_0 is the current when $V_{gs} = V_t$.
- α is a constant dependent on the device fabrication process.

Sub-threshold current depends on fabrication process, temperature variations, and gate voltage.

DIBL *Drain Induced Barrier Lowering* occurs when the drain depletion region interacts with the source near the channel surface, thus lowering the source potential barrier. For short channel devices, when the drain voltage is increased, it lowers the barrier height, resulting in decrease of

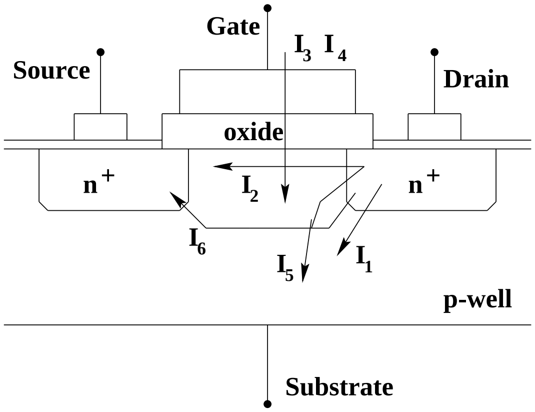


Figure 1.3. Leakage current mechanism in deep-submicron [14].

threshold voltage. The source then injects carriers into the channel without the influence of the gate bias, thus increasing the subthreshold current [14].

3. *Gate Oxide Tunneling* (I_3): As the device size shrinks, there is a corresponding reduction in gate oxide thickness. This process results in a significant increase in the electric field across the gate. The high electric field together with low gate oxide thickness results in tunneling of electrons from the substrate to gate and also from gate to substrate through the gate oxide [14]. The two forms of tunneling across the gate oxide are, namely, *Fowler-Nordheim Tunneling* and *Direct Tunneling.*

4. *Hot Carrier Injection from Substrate to Gate Oxide* (I_4): Short channel devices are susceptible to carrier injection into the gate oxide, due to high electric field near the silicon-oxide interface. Electrons or holes gain sufficient energy to cross the interface barrier and enter the gate oxide. This effect is called as hot-carrier injection [14].

5. *Gate Induced Drain Leakage* (I_5): GIDL is due to the high electric field in the gate/drain overlap region of a transistor. When the MOS is in the accumulation region, surface underneath the gate behaves like a heavily doped region than the substrate, it causes the depletion layer at the surface to be much narrower than elsewhere. This form of leakage occurs at a high drain bias and lower gate bias. The n+ drain region under the gate becomes depleted and sometimes inverted at a low gate bias. For minority carriers the substrate is at a lower potential, hence the carriers caught in the depletion region beneath the gate are swept to the substrate. This effect is known as the

Gate Induced Drain Leakage. Lightly doped drain, high V_{DD} and thin oxide thickness enhance GIDL [14].

6. *Punchthrough* (I_6): In short channel devices, the seperation between the source and drain depletion layers reduces with a corresponding increase in the reverse bias (V_{ds}) voltage. At a sufficient drain voltage, the depletion layers touch or merge deep below the surface causing punchthrough. Since the region near the silicon surface is heavily doped (for a threshold adjust implant), there is a greater expansion of the depletion layer deep below the surface (due to less doping) as compared to the surface. Thus, punchthrough occurs deep below the surface [14].

In this section, we discussed about the two components of power, namely, dynamic and leakage. As we move towards the nano-domain, dynamic component as well as the leakage component of the power is expected to have a steady increase mainly due to the increase in number of transistors per chip. The total power (P_t) expended in a circuit can be expressed as the sum of individual gate power (P_{tg}), which in turn can be broken up into switching and leakage components [61].

$$\begin{aligned} P_t &= \Sigma_g P_{tg} = P_{dg} + P_{sg} \\ &= 0.5\alpha f_{clk} V_{dd}^2 C_{load+wire} + (1-\alpha)\, \Sigma_i P_{leak,i} \beta_i \end{aligned} \tag{1.8}$$

- V_{dd} is the supply voltage.
- f_{clk} is the clock frequency.
- C_{load} is load capacitance.

where α denotes the activity of the node, and β is the probability of remaining in a dominant leakage state (namely signal at 0 or 1). Note that β as well as α is dependent on the switching states of the inputs and the physical parameters. Thus, total power is a function of the input switching states, and the device parameters.

As indicated in [57], dynamic power (indicated as $P_{dynamic}$) will be 90% of the total power if switching activity (average number of switchings in a clock cycle) is more than 0.1. Further from [7], by the year 2005 the leakage component of power, $P_{leakage}$, starts dominating the overall power dissipation. Reliability and performance of a circuit degrades with excessive average power consumption, requiring

Abstraction Level	Speed	Accuracy
Algorithm	High	↑ Worst
Behavior		
RTL		
Logic		
Circuit	↓ Low	Best

Figure 1.4. Accuracy vs. speed of power estimation at various levels [3].

the use of costly packaging and heat sinks to control temperature. Hence, accurate estimation of power during the early design phases such as at transistor, logic, architectural or even behavioral levels will reduce complicated, expensive design changes at later stages due to power dissipation considerations. Thus, modeling *and* estimation of switching activities as well as leakage power remain to be important problems in low-power design. Estimation of power has been performed at various levels of abstraction, namely behavioral level, RT level, gate level, circuit level etc., so that optimization of design can be performed at each level before synthesizing to the lower levels. As depicted in Figure 1.4., higher the level of abstraction, faster is the speed of estimation but with low accuracy. The reason is, at a higher level the designer does not have enough knowledge about the internals of a module, that is, the implementation details of a module. This thesis presents methods to estimate the dynamic component of power as well as leakage component of power at the *gate level*. At this level since we know the logic structure, we can easily estimate the gate capacitance and the major challenge lies in estimating switching activity $(0_{t-1}1_t, 1_{t-1}0_t)$ and the dominant leakage state $(0_{t-1}0_t, 1_{t-1}1_t)$. Later in the same section, we introduce a zero-error compact switching model for power analysis.

The challenges in dynamic power estimation lies in assessing the load capacitance and switching activity since the supply voltage and clock frequency are known to the designers. Switching activity is the number of transistions that a node (input or output) makes per clock cycle. The switching activity of a node is affected by various factors such as the connectivity of the circuit, the input statistics, the correlation between nodes, the gate type, and the gate delays, thus making the estimation process a complex procedure. The correlations among the nodes affect switching activity and it has been observed that models that did not account for node correlations yield less accuracy. There are plethora of techniques available to estimate switching activity namely, simulation, statistical simulation, and

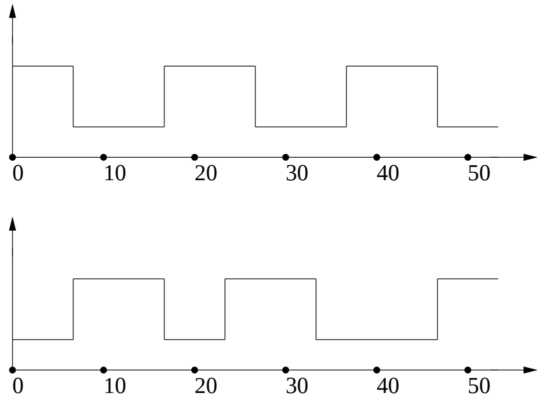


Figure 1.5. Two different signals having identical frequency [3].

probabilistic techniques. Section 1.3 gives a brief introduction to the fundamentals of simulative and probabilistic power analysis techniques.

1.3 Power Analysis Techniques

The idea behind simulation-based approaches is to mimic the circuit behavior over time. Simulation software records the precise time instant at which a signal event occurs. However, simulation techniques are stongly *input pattern dependent.* To completely simulate a circuit for all possible input vectors is impossible as the input vectors depend on the chip in which the module is being placed. Hence, for large circuits estimating switching activity through simualtion is computationally expensive. Note that simulation techniques are accurate, and technology independent. In probability-based approaches, signals are considered as random zero-one process. We no longer know the exact instants at which the logic signal does its transition. Note that to estimate the *frequency* of a signal, there is no need to know the exact time of switching, that is, it is sufficient to record the number of transitions. For example, in Figure 1.5., although the signals appear different, the number of signal events or transitions is identical, which means the frequency of the signals remain the same. Hence, for some purposes (computing power), the two signals in Figure 1.5. remain indistinquishable. Therefore exact characterization of a signal by capturing all its history to study power is inefficient and cumbersome.

The problem of input pattern dependence can be solved if we capture a few essential statistical parameters of a signal. This way we can construct a compact description of a signal and analyze its effect on a circuit. By describing the primary input signals using the characteristic quantities like *sig-*

Table 1.1. Probabilistic switching activity estimation techniques.

Methods	Can it handle			Speed	Accuracy-Time trade-off (any-time aspect)
	Temporal Corr.?	Spatio-Temporal Corr.? *order* $\gg 1$	Input Corr.?		
CREST [27]	Yes	No	Approx.	Fast	No
DENSIM [26]	Yes	No	No	Fast	No
OBDD [28]	Yes	Yes	No	Slow	No
TPS [35]	Yes	No	Approx.	Moderate-Fast	No
Marculescu *et al.* [56]	Yes	No	Approx.	Slow	No
Schneider *et al.* [55]	Yes	No	No	Moderate-slow	No
Marculescu *et al.* [29]	Yes	No	Approx.	Moderate-slow	No
Bhanja *et al.* [5, 24]	Yes	Yes	Approx.	Fast	No
This thesis	Yes	Yes	Approx.	Fast	Yes (zero-error)

nal probability, transition density, etc, and propagating the effects of these quantities to the internal nodes and output of the circuit, we can study the power from the collective influence of all the logic signals. However the propagation of the signals largely depend on the *probabilistic model* used. As stated before, accurate estimation of switching activity requires complete details on signal correlations. But most of the probabilistic models (discussed in Chapter 2) do not consider the correlation (assume temporal and/or spatial independence) among the nodes, as shown in Table 1.1. Hence, they result in inaccurate estimates. It has been established that for zero-delay model of a combinational circuit, only first order temporal correlation is exhibited [54], because signals possess first order Markov property. Thus, it is sufficient to consider just first order temporal correlation, but all high order spatial correlations to model *all* spatio-temporal dependencies in the combinational circuit. In this thesis, we use a probabilistic model using Bayesian Networks, for estimating switching activity, that captures both the first order temporal as well as all high order spatial correlations in a comprehensive manner, resulting in accurate estimates.

1.3.1 Probabilistic Bayesian Network Model

Recently, we proposed a novel model [5, 24], for switching activity estimation in combinational circuits using Probabilistic Bayesian Networks [2], that captures both the temporal and spatial dependencies in a comprehensive manner, resulting in zero-error estimates. Bayesian Networks are a Directed Acyclic Graph (DAG) representations, whose nodes represent random variables and the links denote direct dependencies (capturing the spatial dependencies), quantified by conditional probabilities of a node given the state of its parents. The DAG structure models the joint probability over a set of random variables in a compact manner.

The core idea of this thesis is to express the switching activity of a circuit as a joint probability function which can be mapped, one-to-one, onto a Bayesian Network thus preserving the dependency model of the probability function. We first construct a Logic-Induced-Directed-Acyclic-Graph (LIDAG) based on the logical structure of the circuit. In [24], the author proves that the LIDAG structure, corresponding to the combinational circuit is a minimal representation of the underlying switching dependency model and hence is a Bayesian Network. Each signal in the circuit is considered as a random variable in the LIDAG that can have four possible states indicating the transitions from $0_{t-1} \rightarrow 0_t, 0_{t-1} \rightarrow 1_t, 1_{t-1} \rightarrow 0_t, 1_{t-1} \rightarrow 1_t$. Specification of these four states enables us to completely capture the temporal correlation. Directed edges between the random variables representing switching denotes the probabilistic dependency among the signals. It is our observation that the Bayesian Network is a powerful tool to model switching activity preserving the various dependencies in a circuit. Further, elegant inference mechanisms exist for Bayesian Network computation that make the estimation time-efficient and thus, usable for large circuits.

The attractive feature of this graphical representation of the joint probability function is that it can not only model complex conditional independence over a set of variables, but the independencies serve as a computational scheme for smart probabilistic updating. In general, the belief updating schemes can be classified into exact and approximate techniques. However, the space-time complexity of exact estimation schemes increase with circuit complexity. For instance, the inference scheme that we used in [5, 24], which was a cluster based scheme, resulted exact estimates, however, it was memory intensive. So, for complex circuit, we had to resort to partitioning schemes, resulting in an approximate model of the switching activities in terms of a set of loosely coupled cascaded Bayesian Networks. This model produced estimates with low *mean* error, but due to coupling losses at the boundary nodes, it resulted in larger *standard deviation* and *maximum* error.

From a design point of view, it is sometimes desirable to have an estimation strategy where one can easily trade-off between time and accuracy, essentially an any-time estimation algorithm. This is not possible with the current inference scheme. For these reasons, this work explores a different set of stochastic inference algorithms (approximate BN inference scheme) for three primary reasons:

- As number of transistors increase under nano-domain device shrinkage, losses in the partition would degenerate the estimates even further, and we need a different non-partition based inference mechanism.

- With increased nano-domain complexity, we need an any-time algorithm that should not only generate accurate estimates given enough time, but even under time constraints, should yield rough approximately valid estimates.

- it is not possible to use a partitioned cascaded set of BNs for probabilistic diagnosis in case we want to study the reverse-causal effects, namely the effect of an evidence in an observed node on the primary inputs.

In this thesis, we explore three Stochastic Importance Sampling schemes: Probabilistic Logic Sampling (PLS) [10], Adaptive Importance Sampling (AIS) [11] and Evidence Pre-propagated Importance Sampling (EPIS) [12] based on the zero-error, pure, Bayesian Network switching model for belief updating. These algorithms combine the any-time feature of simulative approaches and input pattern independence of probabilistic approaches. PLS [10], yields excellent estimates when used under predictive situation but in diagnostic reasoning, especially when evidence is unlikely, the accuracy degenerates. However, the predictive mechanism produces the best accuracy-time trade-off. In AIS [11], and EPIS [12] sampling schemes, each sample determines the posterior probability of the underlying model for the remaining samples. The probability of random variable is proven to converge [11] to the correct values given enough time. AIS and EPIS, produce accurate estimates under diagnostic situations, even though the time for predictive mechanism is higher than PLS. Experimental results with ISCAS'85 benchmark shows order of magnitude reduction in maximum error, standard deviation specially for larger benchmarks with significantly low time, especially for the PLS scheme.

Currently, leakage current drawn is not significant enough but is expected to be on par with the dynamic component of power by the year 2005 as the technology moves towards nano dimensions [7]. As the magnitude of leakage current increases it becomes a major contributor to the power consumption, and hence the designers have to face the added burden due to the leakage power dissipation and optimize their designs for low leakage power as well. Note that, it has been found that circuits not only leak during the sleep or idle mode but there is also a significant contribution to leakage power even in active modes [61], [62], [63]. Hence, the designers should also consider the gates that stay dormant during most part of circuit operation while optimizing the circuit for leakage power. Second part of this thesis unveils a new technique for targeting the gates that contribute towards leakage even under the active or run-time mode.

The issue and the approach presented in the second part of the thesis for Leakage Power analysis are new to our knowledge. Our work provides another aresenal for the low power designers to focus on leakage mitigation schemes at targeted nodes, rather than just considering nodes (based on critical path) or using a single (or a handful of data profiles) to target at an optimization scheme. *How do we identify the leaky nodes ?* We exploit the backtracking (reasoning from the evidence to cause) feature of Bayesian Networks to determine the likely input space, given an observation. By using this aspect of Bayesian Network the designer can determine the likelihood of a node being in a leaky state ($0_{t-1}0_t$ or $1_{t-1}1_t$) most of the time, even without any prior knowledge on the input space. Input signals are generally unknown during the design phase because they depend on the system in which the module or chip will be used. Also, it is not possible to simulate the cicuit for all possible input vectors. Designers can exploit the backtracking attribute of Bayesian Networks to determine the transistor that has a high possiblity to leak even in the run-time mode, without prior knowledge of the input space. We use Evidence Pre-propagated Importance Sampling [12] stochastic inference technique to accurately propagate belief from evidence to all other variables. Methods, such as dynamic threshold voltage [59],[60], can use our measure to select target nodes for optimization. We also introduce two entropy based measures to characterize the amount of uncertainities in the posterior input space as an indicator of the leakage of a signal.

1.4 Contributions of this Thesis

1. Bayesian Network models of switching activity are inherently zero-error models. They are *input pattern insensitive.*

2. Bayesian Networks models conditional independencies but does any causal model like BDD. However, the real merit of a BN is that it unifies a graphical model and a probabilistic model invariant in terms of the conditional independencies and the DAG structure is actually a minimal compact I-map of probabilistic model. This probabilistic graphical model uses the advantage of both graph-based and probability-based model for efficient probabilistic updating.

3. The theoretical contribution of our thesis is that the joint probability function of a set of random variables is exactly mapped capturing *all higher order correlations between the signals accurately* using Bayesian Network model. This implies that we can model spatio-temporal correlations of any order (first order temporal is sufficient for zero-delay model) and hence it is an exact switching model. Moreover, we use independence relations not only to model dependencies exactly, but also to use it in our computational advantage during Bayesian inferencing.

4. Bayesian Networks are unique probabilistic causal model in capturing the induced dependence between independent parents of a node given an observed state for the node.

5. Bayesian Networks allow *multi-directional belief flow*. The model can accept evidence from any node and propagate it in any direction.

6. In this thesis, we use non-partition based stochastic inference schemes for switching estimation that

   - scales well.
   - results in anytime estimate (for accuracy-time trade-off).

   We [9], showed that the stochastic inference schemes, namely, Probabilistic Logic Sampling (PLS) [10], Adaptive Importance Sampling (AIS) [11] and Evidence Pre-propagated Importance Sampling (EPIS) [12] result in zero-error estimates for switching activity estimation in combinational circuits. These algorithms combine the any-time feature of simulative approaches and input pattern independence of probabilistic approaches. We achieved an order of magnitude improvement over the past state of the art [5, 24] in terms of maximum error and standard deviation. PLS [10], yields excellent estimates when used under predictive situation while AIS and EPIS, produce accurate estimates under diagnostic situations, even though the time for predictive mechanism is higher than PLS.

7. Bayesian Networks are extremely effective to perform backtracking, the feature of Bayesian Network that is being exploited in the second part of my thesis. Not only can it propagate probabilities from input to the output in a causal flow, it can also propagate probabilities from known evidence or observation to its unknown cause. This is useful and straight-forward for analysis and to resolve queries like "what inputs can cause a switching probability of 0.8 at a particular node of interest?", which would require a search in the input space for other probabilistic or simulative framework.

8. It is impossible to simulate a circuit for all possible input combinations. It depends on the chip in which the module being designed is finally placed. We present a new technique that determines the likely input space of a node, say *X*, given that the node *X* is fixed at one of its states (in our case the possible states of a node are $0 \rightarrow 0,\ 0 \rightarrow 1,\ 1 \rightarrow 0,\ 1 \rightarrow 1$). Chapter 5 exploits the backtracking attribute of Bayesian Networks to target nodes that stay idle even during the runtime mode. We use two entropy based measures (absolute entropy, relative entropy) to quantify the information content of the posterior input space to determine the possiblity of a node to leak. Experimental results with ISCAS'85 benchmark suite show that greater part of the circuit remain dormant especially in active mode and hence, they are the nodes that should be definitely considered while performing leakage power optimization.

1.5 Organization

The rest of the thesis is organized as follows. Chapter 2 contains a literature survey of power estimation techniques. In Chapter 3, we discuss about the fundamentals of Bayesian Networks and the modeling of a combinational circuit into a Bayesian network. We discuss in detail about various probabilistic inference schemes for estimating switching activity in Chapter 4 and conclude the chapter with experimental results on ISCAS benchmark circuits. In Chapter 5, we discuss in detail about two characterization schemes to investigate the likelihood of a node to be at a leaky state even in active mode. We conclude the thesis in Chapter 6.

CHAPTER 2

PRIOR WORK

2.1 Existing Dynamic Power Estimation Techniques

The primary component of logic level power estimation is switching activity estimation. Another important component is capacitance estimation which is simplified due to the knowledge of the circuit structure and the existence of libraries of capacitances for standard cells. The effect of reducing gate delay and growing interconnect delay will also be dominant in future estimation techniques. However, switching activity will remain an important parameter to estimate even in the nano domain mainly due to its dependence on input data and on correlations exhibited in inputs and in the internal nodes.

Definition 1: Switching activity at a node can be defined as the average number of signal transitions in a clock cycle.

In probabilistic terms, $Sw(X)$ is the probability of occurrence of a transition in a node X during a clock period.

$$Sw(X) = P(X_{0\to1}) + P(X_{1\to0}) \tag{2.1}$$

where $X_{0\to1}$ and $X_{1\to0}$ denotes a signal transition from 0 to 1 and 1 to 0 at node X respectively. It is clear that switching activity requires second order statistics. Switching activity $Sw(X)$ can also be denoted as transition probability of X.

Definition 2: The signal probability of a node $P(X = 1)$ is the average fraction of clock cycle that the node X remains at logic 1.

Switching activity of a node is affected by the correlations exhibited. There can be three/four types of correlation. They are

1. Temporal Correlation: This arises since the previous value of a signal can be correlated with the present value of the signal.

2. Spatial Correlation: Spatial correlation arises when the two spatially connected signals are dependent on each other. It is caused in general by re-convergent fan-outs, by feedback and by already correlated primary inputs.

3. Spatio-temporal: Spatio-temporal correlation is dependence of a signal to the previous value of a spatially connected signal. Hence, it is a combination of spatial and temporal correlation.

4. Sequential: It is part of spatial correlation when the dependence is amongst the feedback state lines.

Switching activity can be estimated at various levels of abstraction namely at architectural, behavioral, RTL, logic (discussed extensively in this chapter), or transistor levels. As discussed before, estimation at each level has its own advantages and disadvantages (Figure 1.4.). More power savings can be achieved as power is estimated and optimized at the higher levels, whereas the estimates are significantly more accurate at the transistor level. This work is concentrated at the logic or gate level switching activity estimation. Switching activity estimation strategies can be divided into three broad categories: estimation by simulation, estimation by statistical simulation and estimation by probabilistic techniques.

Estimation by pure simulation [38, 40, 44, 49] though time consuming, is extremely accurate. To decrease the time complexity, several improved simulation techniques have been proposed [33, 39, 41, 42, 43, 48]. Many of them use vector compaction and modeling of input sample space [45, 46, 47] and sequence generation to reduce the samples needed for simulation. The simulation-based techniques are strongly input pattern dependent. In general, all simulation tools have higher accuracy compared to other existing methods but with higher time requirement.

In statistical simulation, statistical methods are applied in conjunction with simulation in order to determine the stopping criterion for the simulation. The earliest works in this area can be found in [25] and [50]. These methods are efficient in terms of the time required and if the statistical distribution of the input data is modeled correctly they can yield accurate estimates. However, one has to be careful in modeling the statistical patterns at the inputs and special attention has to be given to not get trapped in a local minima. These estimation strategies are based on the knowledge of the inputs, the role of input

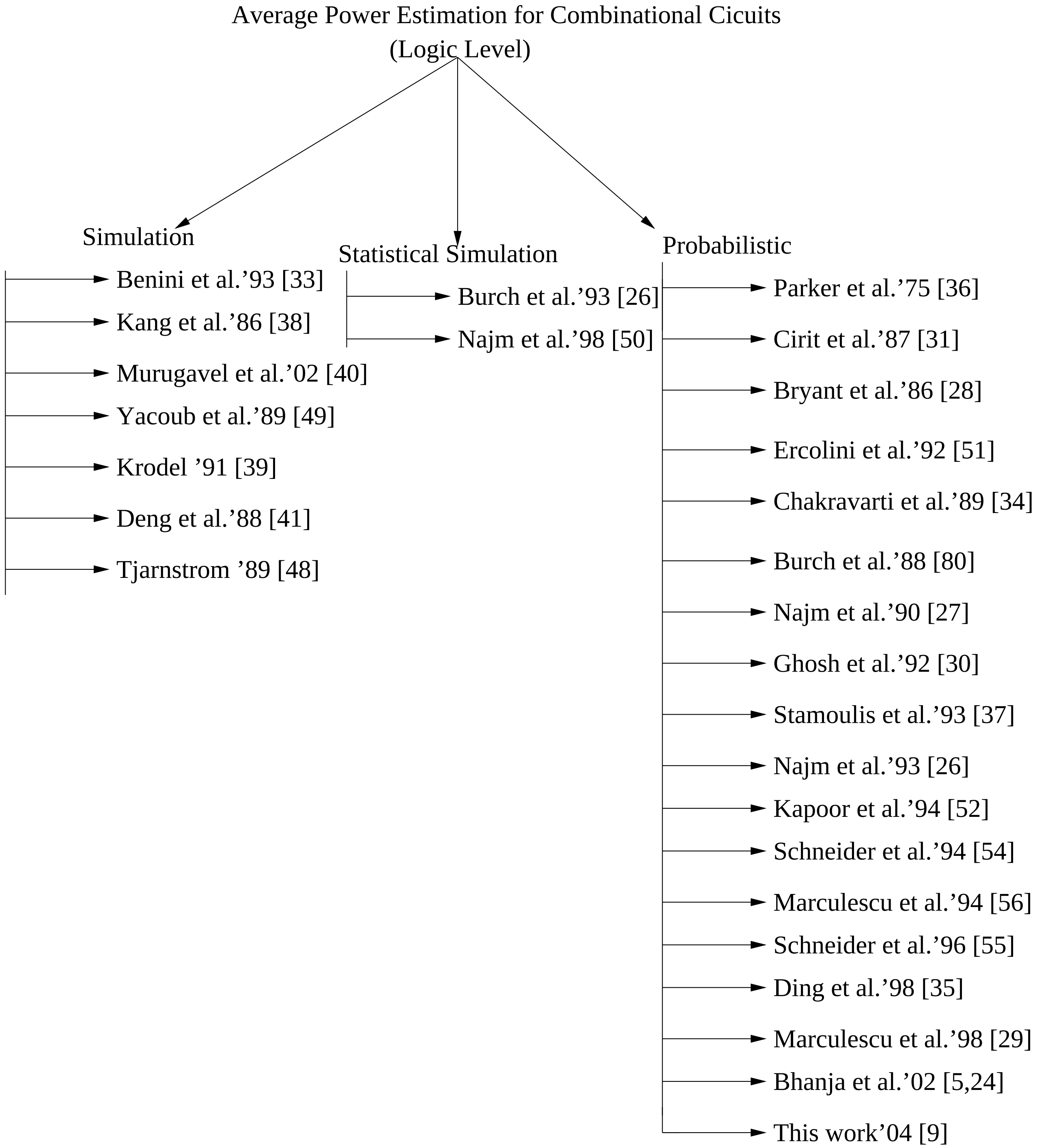


Figure 2.1. Techniques for estimating switching activity in Combinational Circuits.

patterns become very important in terms of whether the sample set represents the entire population or a subset of it.

Probabilistic techniques are fast and more tractable, but typically involve assumptions about joint correlations. The primary conceptual difference is that, the input statistics are first gathered in terms of probabilities and then these probabilities are propagated. Hence, the abstracted knowledge about inputs are used to estimate the switching activity of internal nodes. Hence, these techniques can more easily model changes in input pattern efficiently than other methods. Unlike simulation and statistical simulation, we need to know the dependencies in the circuit structure to propagate probabilities efficiently. Moreover, issues that drastically hamper probabilistic propagation, such as correlations and feedback, have to be modeled accurately. Probabilistic techniques can be further classified into probabilistic simulation [35, 27, 37], and purely probabilistic methods [30, 28, 36].

The earliest effort involved probabilistic propagation of signal probability [31]. Under temporal independence, switching activity can be modeled by signal probability. However, the estimates are grossly inaccurate. Moreover, spatial independence was also assumed.

Most later works use Binary Decision Diagrams (BDD) [28] to compute signal probabilities for all the internal nodes. The use of BDD for signal probability was first proposed by Chakravarti et al. [34]. For most later probabilistic models, BDD is used for probabilistic propagation.

In some of the pioneering works based on probabilistic simulation, Najm *et al.* [27] estimated the mean and variance of current using probability waveforms. It starts with an input probability waveform, which is then propagated throughout the circuit. Probability waveform consists of probability of a signal to be 1 for a certain time interval and probabilities of transition from low to high and from high to low at a particular time instant in the waveform. Najm *et al.* in CREST [27] accounted for temporal correlations during the propagation of the signal and the transition probabilities. However, spatial independence was assumed which resulted in inaccurate estimates, especially in the presence of re-convergent fan-outs.

Najm *et al.* [26] introduced the concept of transition density (denoted by D, a measure of switching activity), which is also a propagation based strategy using so called Boolean difference algorithm.

For simple gates, transition density *D* can be calculated directly whereas one can utilize BDD for complex logic gates. This model can somewhat model the effect of real delay but have reported very high errors due to underlying independence assumptions in the inputs of a function.

In summary, dependency modeling of switching activity has been performed by many of the above methods, but only partially. Present formalisms are not able to account for all types of spatial dependencies. Some of the pioneering works, that enhanced the accuracy of estimation by addressing correlation and dependency issues are discussed in the remaining part of the section.

Kapoor [52] has modeled structural dependencies approximately by partitioning the circuit into local BDDs for signal probability. To improve speed, local BDDs have become increasingly common [29, 35]. Moreover, a partitioning strategy that was followed in [52], tried to maximize the number of correlated nodes in each partition.

Schneider *et al.* [54] used one-lag Markov model to capture temporal dependence. The first order temporal model is valid only under zero delay model, where, the present value of a node is independent of all the past values given just the previous value. This is not the correct picture under real-delay model.

Schneider *et al.* [55] proposed a fast estimation technique based on ROBDD. An approximate solution based on partitioning approach is proposed to attribute reconvergent spatial correlation with reduced time complexity. It is not clear, how accurate this modeling is in terms of the order of spatial correlation.

Modeling spatial correlation using pair-wise correlation between circuit lines was first proposed by Ercolani *et al.* [51]. Tsui *et al.* [53] modeled first order spatial correlation efficiently using correlation coefficients and utilizing them in probabilistic propagation.

Marculescu *et al.* [56], studied temporal, spatial dependencies jointly. In this work, conditional probabilities were used for the lag-one Markov model to capture temporal correlation and the idea of transition correlation coefficient was introduced.

In summary, even the best existing propagations algorithms do not account for higher order spatial correlation. Under zero-delay model, first order temporal effect are sufficient to capture the temporal effects exhibited in the circuit. In fact, in this thesis, the conditional independence relationship is utilized completely and we have computational advantage in modeling accurate higher order spatial correlation propagation.

We tabulate the salient aspects of the works that are most closely related to our thesis, in chapter 1 Table 1.1. to place our thesis in the context of these earlier efforts. Our proposed Bayesian network based formalism is able to handle temporal and spatial correlations in detail. We do not require the inputs to be independent and are able to model correlation among them.

We [5] modeled switching activity, capturing all higher order dependencies, using Bayesian Networks. It has to be noted that conditional independence is used in Bayesian Network to model dependencies, as well as, to construct efficient computational inference schemes. The uniqueness of Bayesian Network based model is that it makes a graphical model and underlying switching model invariant in terms of conditional independence map (the set of all conditional independence relations between any subset of random variables). This makes elegant inference schemes possible based on local message passing. However, they [5] used clustering inference scheme which is an exact algorithm for updating the probabilities of each node. Exact algorithms when applied to networks with large number of nodes requires prohibitive amount of storage and are computation intensive. We employed partitioning scheme [5] to alleviate the problem of complexity. However, losses in the partitions were high and could affect intermediate nodes significantly resulting in high maximum error and standard deviation. The goal of this work is to use approximate stochastic inference schemes for Bayesian Network inference such that estimates are uniformly accurate. These algorithms is a smart combination of the any-time feature of simulative approach and pattern insensitivity of probabilistic approach. Moreover, we achieve an order of magnitude improvement over the past state of the art in terms of maximum error and standard deviation. The theoretical contribution of our thesis is that the joint probability function of a set of random variables is exactly mapped capturing *all higher order correlations between the signals accurately* using Bayesian Network model. This implies that we can model spatio-temporal correlations of any order (first order temporal is sufficient for zero-delay model) and hence it is an exact switching model. Moreover, we use independence relations not only to model dependencies exactly, but also to use it in our computational advantage during Bayesian inferencing.

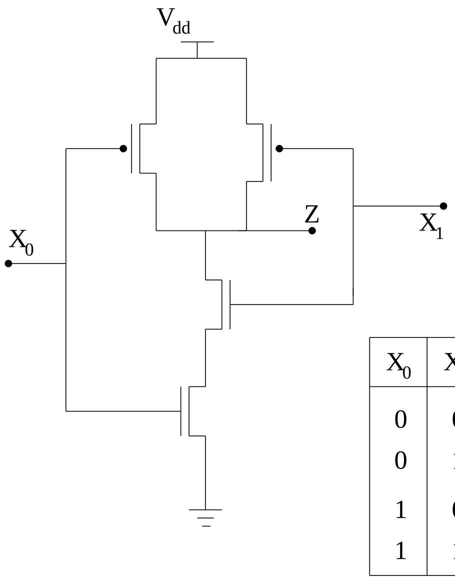
Earlier efforts either treated the distribution as a composition of pair-wise correlated signals between all signals [29, 51] or use an approximate solution for capturing spatial correlation [55]. Moreover, the Bayesian Network models *conditional independence of a subset of signals* unlike in [29]. As a result, complex dependencies exhibited between sets can be modeled. Also, in contrast to [29], the propagation mechanism *does not assume any signal isotropy* of conditional independence.

Further, Bayesian Network based modeling is an unified approach for modeling and propagating probabilities. We can model correlations both in the circuits as well as in the inputs. Our effort in capturing the joint probability distribution function of the whole circuit is unique and has enabled us to model higher order correlations rather than just pair-wise correlations. The propagation algorithms always maintain the overall probability equilibrium of the whole BN and not just between inputs and outputs of a gate.

2.2 Existing Leakage Power Estimation Techniques

As technology scales down, supply voltage must be reduced to keep the dynamic power within its limits. To avoid the negative impact on circuit delay, due to the supply voltage reduction, the threshold voltage of the transistor has to be scaled proportionately. However, scaling of threshold voltage to maintain speed of operation has an adverse effect on leakage, that is, leakage increases as the threshold voltage scales down. Hence designers have to start analyzing and optimizing their design for leakage power as well. One crucial, well-established, observation is that both dynamic and static power are data dependent. Dynamic power is obviously data dependent. Stand-by leakage is also dependent on input states 2.2. [69]. Leakage power optimization techniques have mostly considered stand-by mode [64, 65, 66, 85], and hence are blind to the data-dependence.

The accuracy of leakage power estimation model is dependent on the stand-by leakage current model [82]. Since leakage power depends on the primary input combinations, [82], suggests that the leakage power could be minimized if we apply the input combinations corresponding to the minimum leakage power. To obtain the minimum and maximum values for leakage power dissipation, in [82], the authors developed an accurate leakage model considering the effects of transistor stacks and implemented it in the genetic algorithm framework. Also, in an effort to estimate the proper input combination for minimum leakage vector [83], uses a random search technique to determine low leakage states, without considering the functionality of the circuit. The bounds obtained are not so tight. [84], introduces a new approach for accurate and efficient calculation of the average leakage current in cir-



X_0	X_1	Leakage
0	0	23.60 nA
0	1	47.15 nA
1	0	51.42 nA
1	1	82.94 nA

Figure 2.2. 2-Input Nand Gate and a table showing the dependence of leakage on inputs.

cuits by determining the *dominant leakage states* and use of state probabilites. They also use graph reduction techniques and nonlinear simulation to speedup the simulation time while achieving desired accracy. The author in [86], presents a non-simulative, graph-based algorithms for estimating the maximum leakage power. The algorithm used is pattern independent. The leakage estimation techniques have only considered the stand-by mode leakage. But the gates disspiate leakage power even during the active modes. Hence, for accurate estimation of leakage power we have to consider the leakage occuring in the active or run-time mode as well.

Run time leakage mitigation schemes [67, 68, 69] have also been proposed to dynamically change circuit conditions in response to low leakage high leakage situations. [69] introduces a method to identify the Minimum Leakage Vector(MLV) for leakage power reduction. Since the leakage power depends on the input pattern, the central idea in [69] is to apply miminum-leakage producing input combination to the circuit when it is in the inactive mode, to control the leakage power dissipation. An excellent review on run-time techniques can be obtained in [70], which discusses the issue of limit of leakage reduction and performance penalties associated with the techniques.

Leakage does not just happen during stand-by modes, leakage is also present during active period and is a crucial component of total power optimization [61, 62, 63], however, these are harder to model and handle. Circuits which are in active mode most of the time, or switches frequently between active and stand-by modes will also have nodes that would leak significantly during the active mode. And, this leakage would be data dependent. This component is clearly captured by Nguyen *et al.* [61], where the static component of power during active region is dependent on (1-α) where α is a measure of activity. They proposed a linear programming (LP) based optimization framework for simultaneous assignment of threshold voltage and sizing. They also proposed a dual V_{dd} extension of the problem by ILP formulation. A heuristic is used in [63] for dual Vdd, dual Vth and sizing where they show optimization for three different switching scenario. In [62], the data dependence of the leakage power during active zone is not considered. The optimization criterion used for power optimization is itself data dependent, which make it hard to make generalized statements about the optimality of any operating point found by optimizing it some set of inputs. It is not clear if the optimized values found for one set of input statistics will hold for another set of input statistics. This is a serious concern when designing modular circuits that will be eventually used in different contexts.

In this work, we focus in determining the gates that remains hibernating even in active mode. Detemining these nodes is crucial for leakage power analysis because they contribute significantly to leakage power. Moreover, a chip or module designed for an application might be used in any environment, we might not have any prior knowledge about the input vector streams, that is, during simulation, applied to this module. Simulating the circuit for power estimation would require a *conditional* search of the input space in a simulative framework, which is computationally expensive. Probabilistic inference using Bayesian Networks is the only consistent uncertainty calculus to handle such situations in an efficient manner. We exploit the backtracking aspect of Bayesian Networks to determine the likely input space of any module, given an evidence. We do this by first forcing an internal node in the module to be at a leaky state ($0_{t-1}0_t$ or $1_{t-1}1_t$) and use backtracking to determine the plausible input space (explaination for the evidence) for the internal node to be at the assigned leaky state. In Chapter 5, we present two entropy based measures to characterize this posterior input space to determine the possibility of a node to leak.

CHAPTER 3

MODELING USING BAYESIAN NETWORKS

In this chapter, we introduce the fundamentals of Bayesian Networks and the concept of conditional independence. We focus on two important aspect of Bayesian Networks: dependency modeling, and the notion of minimal representation.

Next, we convert a combinational circuit to a Logic Induced Directed Acyclic Graph (LIDAG) and we then prove that LIDAG is a Bayesian Network for the underlying switching model and hence an exact model for the underlying joint probability distribution of the whole circuit. Any correlation that is present in the joint probability distribution is captured in such a detailed modeling. We discuss our choice of variables, states, edges, and the assignment of conditional probabilities in LIDAG.

3.1 Bayesian Network Fundamentals

Any probability function over a set of random variables $(X_1,\cdots,X_N)$ can be represented as

$$P(X_1,\cdots,X_N) = p(X_n|X_{n-1},X_{n-2},\cdots,X_1)p(X_{n-1}|X_{n-2},X_{n-3},\cdots,X_1)\cdots p(X_1) \tag{3.1}$$

The above expression holds for any ordering of the random variables. This exact representation of the probabilistic knowledge requires encoding of all entries in P$(X_1,\cdots,X_N)$. As the number of random variables increase representation and inference of the probabilistic knowledge based on the above mentioned probabilistic model becomes intractable. The exact representation assumes that every variable is dependent on every other random variable present in the set. It does not take advantage of the conditional independencies present among the variables. Given the state of the parents the condition of the rest of the circuit is irrelavant to the output. For example, the output of a digital gate does not dependent on the condition of the rest of the ciruit given its inputs. This property is called as conditional independence. This leads to the idea to encode the probabilistic knowledge, that is, the joint

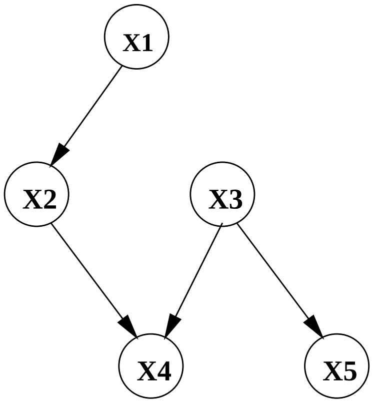


Figure 3.1. Bayesian Graphical Model.

probability distribution of a finite set of variables, concisely using graphical models, which captures conditional independencies embedded among the random variables and arrives at the minimal factored representation in Eq. 3.2, which is a probabilistic model of a Bayesian Network.

$$P(X_1,\cdots,X_N) = \prod_{k=1}^{n} P(X_k|Pa(X_k) \tag{3.2}$$

This form of minimal representation of the joint probability function can be represented as a directed acyclic graph (DAG), with nodes representing the random variables and the links between random variables representing direct probabilistic dependencies.

Let us consider an example which illustrates the concept of conditional independence. In the Figure 3.1., X_4 is dependent on X_2 and X_3 (and X_1 by inheritance). Given X_3, X_4 is conditionally independent of X_5. Let us consider evaluating the joint probability distribution of the random variables X_1,X_2,X_3,X_4,X_5. Using the exact representation,

$$P(X_1,X_2,X_3,X_4,X_5) = P(X_5|X_4,X_3,X_2,X_1)P(X_4|X_3,X_2,X_1)P(X_3|X_2,X_1)P(X_2|X_1)P(X_1) \tag{3.3}$$

The conservative assumption that every random variable is dependent on every other random variable makes the representation and updating of the probabilistic knowledge using the exact model inefficient. Bayesian Networks resolves this issue by exploiting the conditional independence among random variables. Hence by Figure 3.1., the joint probability function factorizes to

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_5|X_3)P(X_4|X_3, X_2)P(X_3)P(X_2|X_1)P(X_1) \tag{3.4}$$

The attractive feature of this graphical representation of the joint probability function is that it can not only model complex conditional independence over a set of variables, but the independencies serve as a computational scheme for smart and efficient probabilistic updating.

3.2 Mathematical Formalism

In this section, we discuss the fundamental modeling issues relevant to Bayesian Network. Interested reader is recommended to read [2] for detailed understanding. As we mentioned before, Bayesian Networks are compact graphical probabilistic model for the underlying joint probability distribution function. Each node in the DAG structure is a random variable representing switching and can have four states ($0 \to 0, 0 \to 1, 1 \to 0, 1 \to 1$) for complete capture of temporal dependence under zero-delay scenario. Edges in the DAG denotes cause and effect relationship in the probabilistic model and is quantified by the conditional probability of a child node given its parents.

To formalize the concept of dependencies, we first present the concept of conditional independence. We begin with the definition of *conditional* independence among three sets of random variables.

Definition 1: Let U=$\{U_1, U_2, \cdots U_n\}$ be a finite set of variables that can assume discrete values. Let $P(.)$ be the joint probability function over the variables in U, and let X, Y and Z be any three subsets of U. X, Y and Z may or may not be disjoint. X and Y are said to be *conditionally independent* given Z if

$$P(x|y,z) = P(x|z) \text{ whenever } P(y,z) > 0 \tag{3.5}$$

Following Pearl [2], this conditional independence amongst X, Y, and Z is denoted as $I(X,Z,Y)$ in which X and Y are said to be *conditionally independent* given Z. Conditional independence implies that knowledge of Z makes X and Y independent of each other. In Figure 3.2. for example, let us denote the switching activity at line i by random variable X_i. U is defined as a set=$\{X_1, \cdots X_8\}$. Switching in a combinational circuit follows directed Markov property, that is, the output of a gate is dependent only

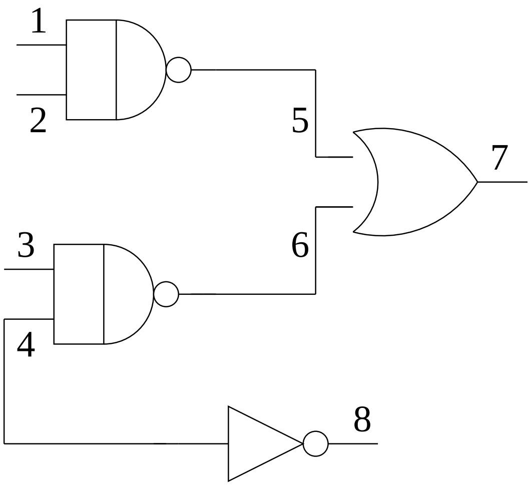


Figure 3.2. A combinational circuit.

on its inputs. Thus the random variable X_7 is completely independent of X_4 given $\{X_5, X_6\}$. Hence, I($X_7, \{X_5, X_6\}, X_4$) is one of the many independencies that are present in the circuit.

A dependency model, *M*, of a domain should capture all these triplets namely (X, Z, Y) conditional independencies amongst the variables in that domain. The joint probability density function is one such dependency model. The properties involving the notion of independence are axiomatized by the following theorem.

Theorem 1: Let X, Y, and Z be three distinct subsets of U. If $I(X, Z, Y)$ stands for the relation "X is independent of Y given Z" in some probabilistic model *P*, then *I* must satisfy the following four independent conditions:

$$I(X, Z, Y) \Rightarrow I(Y, Z, X) \quad \text{(symmetry)} \tag{3.6}$$

$$I(X, Z, Y \cup W) \Rightarrow I(X, Z, Y) \& (X, Z, W) \quad \text{(decomposition)} \tag{3.7}$$

$$I(X, Z, Y \cup W) \Rightarrow I(X, Z \cup W, Y) \quad \text{(weak union)} \tag{3.8}$$

$$I(X, Z, Y) \& I(X, Z \cup Y, W) \Rightarrow I(X, Z, Y \cup W) \quad \text{(contraction)} \tag{3.9}$$

Proof: For proof, see [13].

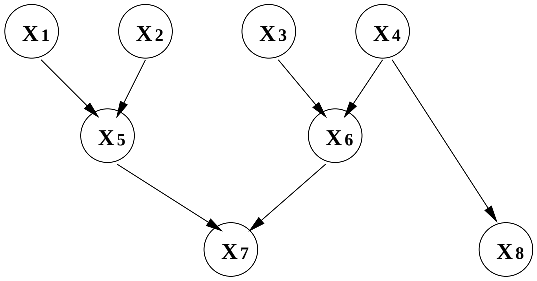


Figure 3.3. Bayesian network corresponding to the circuit in Figure 3.2..

Next, we introduce the concept of *d-separation* of variables in a directed acyclic graph structure (DAG), which is the underlying structure of a Bayesian network. This notion of *d-separation* is then related to the notion of independence amongst triple subsets of a domain.

Definition 2: If X, Y, and Z are three distinct node subsets in a DAG D, then X is said to be *d-separated* from Y by Z, $< X|Z|Y >$, if there is no path between any node in X and any node in Y along which the following two conditions hold: (1) every node on the path with converging arrows is in Z or has a descendent in Z and (2) every other node is outside Z. If there exist such a path where the above two conditions hold, the path is called an active path.

Consider the example DAG in Figure 3.3., let $X = \{X_5\}$, $Y = \{X_6\}$ and $Z = \{X_7\}$. Path $X_5 \rightarrow X_7 \leftarrow X_6$ is active since given information on node X_7, X_5 and X_6 are not d-seperated. It is worth mentioning that if any one path is active, even though the other paths are blocked, the nodes are not d-separated. In the same example, X_7 is d-separated from X_2 by X_5 since the only path $X_2 \rightarrow X_5 \rightarrow X_7$ is blocked.

Definition 3: A DAG D is said to be an I-map of a dependency model M if every *d-separation condition* displayed in D corresponds to a valid conditional independence relationship in M, i.e., if for every three disjoint sets of vertices X, Y, and Z, we have, $< X|Z|Y > \Rightarrow I(X,Z,Y)$. In Figure 3.3., for example $< X_7|X_5|X_1 >$ implies the independence relation $I(X_7,X_5,X_1)$ in the dependency model M formed by the random variables depicting switching activity at a line in the combinational circuit in Figure 3.2..

Note that the Definition 3 holds the unifying feature of the graph based probability model in a way that connects the DAG D to the probabilistic model P. In Bayesian Networks, we not only suggest that DAG D is a dependency model for P (because all the d-separations in D imply a conditional independence in P), but also the notion of a compact minimal representation is built in. Let us consider

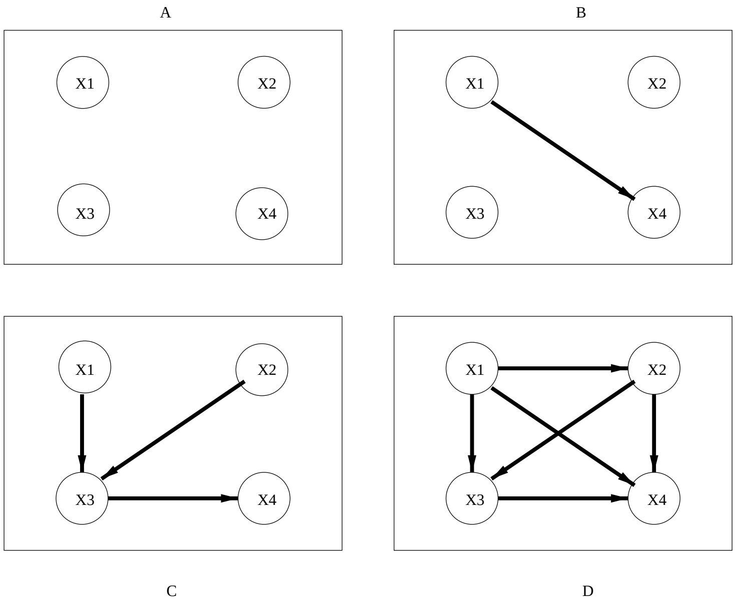


Figure 3.4. Bayesian Networks: Marriage between Graphical and Probabilistic Models.

the example of a probabilistic model P over four random variables $\{X_1, X_2, X_3,\ and\ X_4\}$ as shown in Figure 3.4.. Note that, the DAG in Figure 3.4.a, all the nodes are considered independent and hence I-map of D is greater than that of P which indicates that D under-represents P. In Figure 3.4.d, the I-map of D is less than that of P as D is a complete DAG exhibiting maximum dependencies. This model would generate accurate results but are over-representation and hence the computation efforts would be large. A Bayesian Network has to be the DAG where the I-map for DAG matches the I-map of the P and hence it is the exact representation that is minimal in structure.

Eq. 3.1 denotes the exact probabilistic model over random variables and using conditional independencies (in Eq. 3.10), we can arrive at the minimal factored representation shown in Eq. 3.2 which is the probabilistic model of Bayesian Network.

$$p(x_i|x_{i-1}, x_{i-2}, \cdots, x_1) = p(x_i|Pa(x_i)) \tag{3.10}$$

Definition 4: A DAG is a *minimal* I-map of M if none of its edges can be deleted without destroying its dependency model M.

Definition 5: Given a probability function P on a set of variables U, a DAG D is called a *Bayesian Network* of P if D is a minimum I-map of P.

In general, it is hard to find all the I-maps given a probability distribution function or a graphical representation. There is an elegant method of inferring the minimal I-map of P that is based on the notion of a Markov blanket and a boundary DAG, which are defined next.

Definition 6: A Markov blanket of element $X_i \in U$ is a subset S of U for which $I(X_i, S, U - S - X_i)$ and $X_i \notin S$. A set is called a Markov *boundary*, B_i of X_i if it is a minimal Markov blanket of X_i, i.e., none of its proper subsets satisfy the triplet independence relation.

Definition 7: Let M be a dependency model defined on a set $U = \{X_1, \cdots, X_n\}$ of elements, and let d be an ordering $\{X_{d1}, X_{d2}, \cdots\}$ of the elements of U. The *boundary strata* of M termed as B_M relative to d is an ordered set of subsets of U, $\{B_{d1}, B_{d2}, \cdots\}$ such that each B_{di} is a Markov boundary (defined above) of X_{di} with respect to the set $U_{di}(\subset U) = \{X_{d1}, X_{d2}, \cdots, X_{d(i-1)}\}$, i.e. B_{di} is the minimal set satisfying $B_{di} \subset U$ and $I(X_{di}, B_{di}, U_{di} - B_{di})$. *The DAG created by designating each B_{di} as the parents of the corresponding vertex X_{di} is called a* boundary DAG *of M relative to d.* It should be noted here that the only ordering restriction is that the variables in the Markov Boundary set (of a particular variable) have to be ordered before the random variable.

This leads us to the final theorem that relates the Bayesian network to I-maps, which has been proven in [2]. This theorem is the key to constructing a Bayesian network.

Theorem 2: Let M be any dependency model satisfying the axioms of independence listed in Eqs. 3.6-3.9. If the graph structure D is a boundary DAG of M relative to ordering d, then D is a minimal I-map of M.

Proof: For proof, see [13].

This theorem along with definitions 2, 3, and 4 above, specifies the structure of the Bayesian network. We use these to prove our theorem regarding the structure of Bayesian network to capture the switching activity of a combinational circuit.

Let a combinational circuit consist of gates $\{G_1, \cdots, G_N\}$ with n primary input signals denoted by the set $\{I_1, \cdots, I_n\}$. Let the output of gate G_i be denoted by O_i. The inputs to a gate are either a primary input signal or output of another gate. The switching of these input signal and output lines, $\{I_1, \cdots, I_n, O_1, \cdots, O_N\}$, are the random variables of interest. Note that the set of output lines include both intermediate lines and the final output lines. Let X_i be the switching at the i-th line, which is either an input or an output line, taking on four possible values, $\{x_{00}, x_{01}, x_{10}, x_{11}\}$, corresponding to the possible transitions: $0 \to 0, 0 \to 1, 1 \to 0$, and $1 \to 1$.

Definition 8: A Logic Induced Directed Acyclic Graph (LIDAG) structure, *LD*, corresponding to a combinational circuit consists of nodes, X_is, representing the switching at each line and links between them is constructed as follows: The parents of a random variable representing the switching at an output line, O_i, of a gate G_i are the nodes representing switchings at the input lines of that gate. Each input line is either one of $\{I_1, \cdots, I_n\}$ or an output of another gate. The DAG shown in Figure 3.3. is a LIDAG corresponding to the combinational circuit shown in Figure 3.2..

Theorem 3: The LIDAG structure, *LD*, corresponding to the combinational circuit is a minimal I-map of the underlying switching dependency model and hence is a Bayesian network.

It is interesting to note that the LIDAG structure corresponds exactly to the DAG structure one would arrive by considering the *principle of causality*, which states that one can arrive at the appropriate Bayesian network structure by directing links from nodes that represent causes to nodes that represent immediate effects [2]. Thus, directed links in the graph denote immediate cause and effect relationship. In a combinational circuit the immediate causes of switching at a line are the switchings at the input lines of the corresponding gate.

3.3 Formation of the LIDAG-BN

We first illustrate with an example how switching in a combinational circuit at circuit level can be represented by a LIDAG structured Bayesian network (LIDAG-BN). Then we show how the conditional probabilities that quantify the links of LIDAG-BN are specified.

Let us consider the circuit with fives gates shown in Figure 3.2. We are interested in the switching at each of the 8 numbered lines in the circuit. Each line can take four values corresponding to the four possible transitions: $\{x_{00}, x_{01}, x_{10}, x_{11}\}$. Note that this way of formulating the random variable effectively models temporal correlation since only first order temporal correlation is exhibited in combinational circuit under zero-delay scenario [54]. To capture all higher order spatial correlations, we form the interconnection (through edges) and quantify them by the conditional probabilities for the child-parent group in the LIDAG. The probability of switching at a line would be given by

$P(X_i = x_{01}) + P(X_i = x_{10})$[1]. The LIDAG structure for the circuit is shown in Figure 3.3.. Dependence among the nodes that are not connected directly is implicit in the network structures. For example, nodes X_1 and X_2 are independent of each other, however, they are *conditionally* dependent given the value of say node X_5. Or the transition at line 5, X_5, is dependent on the transitions at lines 1 and 2, represented by the random variables, X_1 and X_2, respectively. Thus, the transitions of line 5 are *conditionally independent* of all transitions at other lines *given* the transition states of lines 1 and 2.

For the Bayesian network structure in Figure. 3.3. the corresponding joint probability density is given by the following factored form. It has to be noted that this factored form can only be obtained for circuits without a feedback.

$$P(x_1, \cdots, x_8) = \quad P(x_8|x_4)P(x_7|x_5,x_6)P(x_6|x_3,x_4)P(x_5|x_1,x_2)P(x_4)P(x_3)P(x_2)P(x_1) \tag{3.11}$$

The conditional probabilities of the lines that are directly connected by a gate can be obtained knowing the type of the gate. For example, $P(X_5 = x_{01}|X_1 = x_{01}, X_2 = x_{00})$ will be always 1 because if one of the inputs of an OR gate makes a transition from 0 to 1 and the other stays at 0 then the output always makes a transition from 0 to 1. A complete specification of the conditional probability of $P(x_5|x_1,x_2)$ will have 4^3 entries since each variable has 4 states. These conditional probability specifications are determined by the gate type. Thus, for a NAND gate, if one input switches from 0 to 1 and the other from 1 to 0, the output remains at 1. We describe the conditional probability specification for a two input NAND and a two input OR gate in Table 3.1. and in Table 3.2. respectively. By specifying a detailed conditional probability we ensure that the spatio-temporal effect (first order temporal and higher order spatial) of any node are effectively modeled.

The last four terms in the right hand side of Eq. 3.11 represent the statistics of the input lines. Given the statistics of the input lines, we would like to infer the probabilities of all the other nodes. A brute force way of achieving this would be to compute the marginal probabilities by summing over possible states, thus, $P(x_8, x_1) = \sum_{x_2,\cdots,x_7} P(x_1, \cdots, x_9)$. This, obviously, is computationally very expensive and, in addition, does not scale well. In the next chapter, we show how the structure of the Bayesian network can be used to efficiently compute the required probabilities.

[1] Probability of the event $X_i = x_i$ will be denoted simply by $P(x_i)$ or by $P(X_i = x_i)$.

Table 3.1. Conditional probability specifications for the output and the input line transitions for two input NAND gate.

Two Input NAND gate					
$P(X_{output} \mid X_{input1}, X_{input2})$ for $X_{output} =$				X_{input1} =	X_{input2} =
$\{x_{00}$	x_{01}	x_{10}	$x_{11}\}$		
0	0	0	1	x_{00}	x_{00}
0	0	0	1	x_{00}	x_{01}
0	0	0	1	x_{00}	x_{10}
0	0	0	1	x_{00}	x_{11}
0	0	0	1	x_{01}	x_{00}
0	0	1	0	x_{01}	x_{01}
0	0	0	1	x_{01}	x_{10}
0	0	1	0	x_{01}	x_{11}
0	0	0	1	x_{10}	x_{00}
0	0	0	1	x_{10}	x_{01}
0	1	0	0	x_{10}	x_{10}
0	1	0	0	x_{10}	x_{11}
0	0	0	1	x_{11}	x_{00}
0	0	1	0	x_{11}	x_{01}
0	1	0	0	x_{11}	x_{10}
1	0	0	0	x_{11}	x_{11}

Table 3.2. Conditional probability specifications for the output and the input line transitions for two input OR gate.

Two Input OR gate					
$P(X_{output}\|X_{input1}, X_{input2})$ for $X_{output} =$				X_{input1} =	X_{input2} =
$\{x_{00}$	x_{01}	x_{10}	$x_{11}\}$		
1	0	0	0	x_{00}	x_{00}
0	1	0	0	x_{00}	x_{01}
0	0	1	0	x_{00}	x_{10}
0	0	0	1	x_{00}	x_{11}
0	1	0	0	x_{01}	x_{00}
0	1	0	0	x_{01}	x_{01}
0	0	0	1	x_{01}	x_{10}
0	0	0	1	x_{01}	x_{11}
0	0	1	0	x_{10}	x_{00}
0	0	0	1	x_{10}	x_{01}
0	0	1	0	x_{10}	x_{10}
0	0	0	1	x_{10}	x_{11}
0	0	0	1	x_{11}	x_{00}
0	0	0	1	x_{11}	x_{01}
0	0	0	1	x_{11}	x_{10}
0	0	0	1	x_{11}	x_{11}

CHAPTER 4

BAYESIAN INFERENCE FOR SWITCHING ACTIVITY ESTIMATION

In the previous chapter, we proved that the Bayesian Network models the exponentially sized joint probability distribution in a compact manner by exploiting the conditional independence relationships present among the random variables. The attractive feature of this graphical representation of the joint probability function is that it can not only model complex conditional independence over a set of variables, but the independencies serve as a computational scheme for smart probabilistic updating. This chapter starts with a quick introduction to ”what is probabilistic inference ?”, and the two important inference schemes used to estimate the beliefs or probabilities in a Bayesian Network. Major part of this chapter is devoted towards exploring different set of stochastic importance sampling (approximate Bayesian inference scheme) schemes namely, Probabilistic Logic Sampling [10], Adaptive Importance Sampling (AIS) [11] and Evidence pre-propagated Importance sampling [12] for Bayesian inferencing. We conclude this chapter with experimental results on ISCAS benchmark circuits.

4.1 Probabilistic Inference

Probabilistic inference or commonly referred as belief updating amounts to calculating the probability distribution of a query node given an observation or evidence. This amounts to computing $P(X|E)$ (bayes theorem).

$$P(X|E=e) = \frac{P(X/E, E=e)}{P(E=e)} \tag{4.1}$$

Computation of the probability of evidence is P(E=e) requires summation over all the variables in the set except the evidence variables and this is expressed in equation 4.2.

$$P(E=e) = \sum_{X/E} P(X/E, E=e) \tag{4.2}$$

For small networks, computation of the probability of evidence using the exact representation that is, equation 4.2 is simple. However, as network size increases computation of $P(E = e)$ efficiently becomes a computational complex problem. Different schemes to process the equation 4.2 give different inference algorithms. Two important Bayesian Network inference algorithms are 1) exact inference, 2) approximate inference. Exact inference algorithms like clustering, pearl's polytree algorithm etc. provide exact estimate [13, 23], however for very large networks they stumble due to NP-hardness of inference [16]. Exact inference applied on large networks are either storage intensive or computationally extensive. To resolve this issue approximate inference methods like Model Simplification, Search based, Loopy belief propagation, stochastic importance sampling were developed. While approximate inference is proved to be NP-hard as well [17], it is the only alternative way to arrive at an estimate for large and complex ciruits. A prominent subclass of approximate inference algorithms are stochastic sampling algorithms. Some instances of these are Probabilistic Logic Sampling [10], Likelihood weighting [18, 19], backward sampling [20], and importance sampling [19]. In this chapter, we explore three important stochastic importance sampling schemes: Probabilistic Logic Sampling [10], Adaptive Importance Sampling (AIS) [11] and Evidence pre-propagated Importance sampling [12] for Bayesian inferencing. These algorithms combine the any-time feature of simulative approaches and input pattern independence of probabilistic approaches.

4.2 Stochastic Inference Algorithms for Switching Activity Estimation

Stochastic sampling algorithms are approximate BN inference schemes. Probabilities are inferred by a complete set of samples or instantiations that are generated for each node in the network according to the importance conditional probability distribution of this node given the values of the parents. In these sampling schemes, each sample determines the posterior probability of the underlying model for the remaining samples. The probability of a random variable is proven to converge [11] to its correct value given enough time. The salient features of these algorithms are: (1) They scale extremely well for larger systems making them a target inference for nano-domain billion transistor scenario (2) They are any-time algorithm, providing adequate accuracy-time trade-off and (3) The samples are not based on inputs and the approach is input pattern insensitive. The classes of algorithms selected here are known as Importance Sampling algorithms [11, 12, 10] which are not only good predictors or estimators, predicting the behaviors of descendent nodes (intermediate ones) given some properties of the primary inputs, but also accurate diagnostic tool that would provide possible pattern of the inputs given a particular set of behavior (evidence) on any internal nodes.

Before we review various stochastic inference methods, it is extremely necessary to understand the theory behind importance sampling, which acts as the backbone for these stochastic inference methods. Readers interested in more details are directed to the literature on Monte carlo methods in finite integrals computation [58]. Let us consider the approximate computation of the integral J,

$$J = \int_{\Theta} b(X)dX \tag{4.3}$$

Let $b(X)$ be a function of k variables $X = (X_1, X_2, ...X_k)$ over a domain $\Theta \subset R^k$. The integral in equation 4.3 can be solved by numerical integartion techniques like Trapezoidal rule, Simpson's rule, Monte carlo method. The integral in equation 4.3 is usually computed by means of either Trapezoidal rule or Simpson's rule as they arrive at precise estimates. But for multivalued integrals, use of the above methods is computationally intensive and hence we resort to Monte carlo method. In this section, we will use Monte carlo based technique for the approximate evaluation of the integral J. Monte carlo methods use random numbers to perform numerical integration. For accuarcy, sampling of random numbers is performed on a distribution that is a reasonable approximation to the actual function $b(X)$. Hence, we introduce an arbitrary density function also called as the importance function $i(X)$ in this integral, $J = \int_{\Theta} \frac{b(X)}{i(X)} i(X)dX$. The importance function $i(X)$ is a probability density function such that $i(X) > 0$ for any $X \subset \Theta$. After sampling the importance function over M instantiations $X_1, X_2, ...X_M$, the approximate value of the integral is calculated as follows,

$$\hat{(}J) = \frac{1}{M} \sum_{i=1}^{M} \frac{b(X_i)}{i(X_i)} \tag{4.4}$$

For large values of M, the distribution of $i(X)$ approaches the distribution of $b(X)$ and hence the the accuracy of $\hat{(}J)$ increases. The variance between the two distributions is minimized when $i(X)$ is proportional to $|b(X)|$. The main goal of the Importance Sampling algorithm is achieving the importance function. While AIS arrives at an importance function by learning from the samples generated during each iteration, EPIS arrives at an importance function by using yet another approximate inference

scheme called the Loopy Belief Propagation. Note that, it is possible to use $b(X)$ as a guide in choosing $i(X)$. The reason why importance sampling technique is used becomes evident when we compare the equation 4.2 and equation 4.3 and conclude that they are almost identical except for the integration which is replaced by summation and the domain Θ is replaced by X
E. The stochastic sampling strategy works because in a Bayesian Network the product of the conditional probability functions for all nodes is the optimal importance function.

4.2.1 Probabilistic Logic Sampling

Probabilistic Logic Sampling (PLS) is the first and the simplest sampling algorithms proposed for Bayesian Networks [10]. The flow of the algorithm is as follows:

1. Complete set of samples are generated for the Bayesian Network using the importance function, which is initialized to joint probability function $P(X)$. The importance function is never updated once its initialized. Without evidence, $P(X)$ is the optimal importance function for the evidence set.
2. Samples that are incompatible with the evidence set are discarded.
3. The probability of all the query nodes are estimated based on counting the frequency with which the relevant events occur in the sample. In predictive inference, logic sampling generates precise values for all the query nodes based on their frequency of occurrence but with diagnostic reasoning, this system fails to provide accurate estimates because of large variance between the optimal importance function and the actual importance function used. The disadvantage of this approach is that in case of unlikely evidence, we have to discard most samples and thus the performance of the PLS approach deteriorates.

4.2.2 Adaptive Importance Sampling

Our objective is estimating the probability of evidence $P(E = e)$. The posterior probability is given by equation 4.1. The optimal importance function for calculating $P(E = e)$ is $P(X|E = e)$ [11]. Although we know the mathematical expression for the optimal importance function, it is computationally expensive to obtain this function exactly. By exploiting the structural advantage of Bayesian Network (the joint probability function that is modeled by a BN can be expressed as the product of the conditional probability of the nodes given its parent nodes) we can arrive at an approximate importance function. The approximate importance function is given as,

$$\rho(X/E) = \prod_{k=1}^{m} P(X_k|Pa(X_k, E)) \tag{4.5}$$

This function considers the effect of evidence on rest of the circuit. $P(E = e) = \frac{P(X/E, E=e)}{\rho(X/E)}$ is an estimate of the probability of evidence (P(E=e)). The equation 4.5 is the importance conditional probability table (ICPT) of a node X and it represents the table of posterior probabilities. The ICPT table will be updated based on the samples at various stages. A significant saving in computation time is reported in [11] if the nodes that are not the ancestors of the evidence nodes are not considered during learning. Hence ICPT of the nodes that are not the ancestors of the evidence nodes are equal to their CPT throughout the learning process.

The steps for this algorithm are presented below:

1. The nodes are arranged in topological order. Each evidence node is instantiated to its observed state and is omitted from further sample generations. Each root node is randomly instantiated to one of its possible states according to the importance prior probability of the node.
2. Each node whose parents were already instantiated will be instantiated to one of its possible outcomes, according to its importance conditional probability table, which can be derived from the importance function.
3. Conditional probability of the evidence set given the sample instantiation is calculated and stored and used to update the importance function after a few run by applying Bayesian Network learning algorithms. This function will then be used for the next stage of sampling. The posterior probabilities are then calculated from the samples.

4.2.3 Hybrid Scheme

For large circuits, a hybrid scheme, specifically the Evidence Pre-propagated Importance Sampling (EPIS) [12], which uses local message passing and stochastic sampling, is appropriate. This method scales well with circuit size and is proven to converge to correct estimates. These classes of algorithms are also anytime-algorithms since they can be stopped at any point of time to produce estimates. Of course, the accuracy of estimates increases with time.

The EPIS algorithm is based on Importance Sampling that generates sample instantiations of the *whole* DAG network, i.e. all for line switching in our case. These samples are then used to form the final estimates. This sampling is done according to an importance function. In a Bayesian Network, the product of the conditional probability functions at all nodes form the optimal importance function. Let $X = \{X_1, X_2, \cdots, X_m\}$ be the set of variables in a Bayesian Network, $Pa(X_k)$ be the parents of X_k, and E be the evidence set. Then, the optimal importance function is given by

$$P(X|E) = \prod_{k=1}^{m} P(X_k|Pa(X_k,E)) \tag{4.6}$$

This importance function can be approximated as

$$P(X|E) = \prod_{k=1}^{m} \alpha(Pa(X_k))P(x_k|Pa(X_k))\lambda(X_k) \tag{4.7}$$

where $\alpha(Pa(X_k))$ is a normalizing constant dependent on $Pa(X_k)$ and $\lambda(X_k) = P(E^-|x_k)$, with E^+ and E^- being the evidence from parent set and child set, respectively, as defined by the directed link structure. Calculation of λ is computationally expensive and for this, Loopy Belief Propagation (LBP) [21] over the Markov blanket of the node is used. Yuan *et al.* [12] proved that for a poly-tree, the local loopy belief propagation is optimal. The importance function can be further approximated by replacing small probabilities with a specific cutoff value.

4.2.3.1 Loopy Belief Propagation

In this part, we would outline Pearl's [2], [81] distributed local message passing scheme that allows efficient backtracking in poly-tree and show an approximation of the poly-tree called loopy belief propagation (LBP) [21] which extends to network with loop (re-convergence) for many applications.

Here, we briefly summarize Pearl's belief propagation algorithm on the Figure 4.1. that is a poly-tree. Each node X computes its posterior probability based on the information obtained from its neighbors. i.e., $Bel(x) = P(X = x|E)$, where E represents the evidence set. In a poly-tree, any node X d-separates E into 2 subsets, E_x^+ which is the evidence connected to node X through its parents Z and

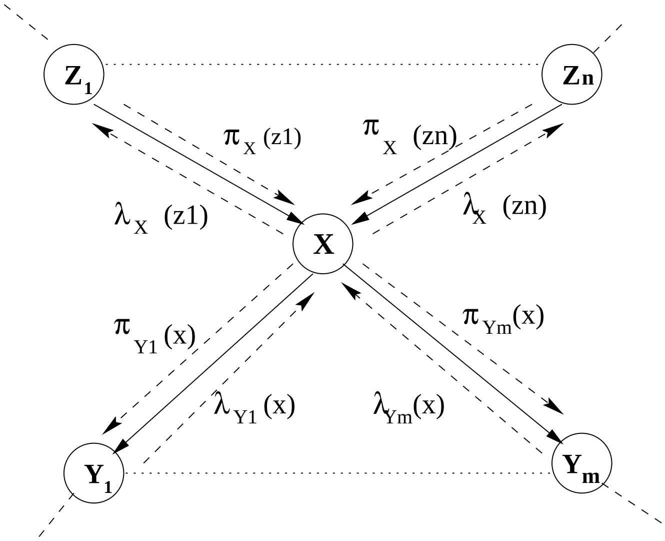


Figure 4.1. Probabilistic inference using local message passing.

E_x^- is the evidence connected to node X through its children Y. Now, the node X can compute its belief by separately combining the messages obtained from its parents and children.

$$Bel(x) = \alpha\lambda(x)\pi(x) \tag{4.8}$$

where $\lambda(x)$ and $\pi(x)$ are given by

$$\lambda(x) = \prod_U \lambda_U(x) \tag{4.9}$$

U is a set containing all children of X.

$$\pi(x) = \sum_{z_1,z_2,\cdots,z_n} (P(x|z_1,z_2,\cdots,z_n) \prod_{i=1}^{n} \pi_X(z_i)) \tag{4.10}$$

where $Z_1, Z_2, \cdots, Z_n$ are parents of node X.

Once the node computes its belief it propagates the updated messages to its neighbors and this iteration is carried until the convergence of the posteriors. The message to the parent Z_l of node X is given by:

$$\lambda_X(z_l) = \sum_x (\sum_{z_1,z_2,\cdots,z_k} (P(x|z_l,z_1,z_2,\cdots,z_k) \prod_{i=1,i\neq l}^{k} \pi_X(z_i)))\lambda(x) \tag{4.11}$$

where $Z_1, Z_2, ..., Z_l, \cdots Z_k$ are other parents of X. The message from node X to its child is given by:

$$\pi_Y(x) = \pi(x) \prod_{C \in CH_X - Y} \lambda_C(x); \tag{4.12}$$

Pearl's belief propagation algorithm can be applied to networks with loops where the belief of a node is continuously updated in a loop till belief has converged. Many applications have shown enormous success and correct convergence using LBP. In [21] it is shown to be connected with the Kikuchi approximation of variational Bethe free energy in statistical physics. Note that, we are not using LBP directly, we use LBP to arrive at an importance function for stochastic inference discussed in the previous section. Importance sampling algorithms have been shown [11] to converge correctly even when the importance function varies slightly from the optimum one.

The above set of stochastic sampling strategies discussed in subsection 4.2.1, 4.2.2, and 4.2.3 work because in a Bayesian Network the product of the conditional probability functions for all nodes is the optimal importance function. Because of this optimality, the demand on samples is low. We have found that just thousand samples are sufficient to arrive at good estimates for the ISCAS85 benchmark circuits. *Note that this sampling based probabilistic inference is non-simulative and is different from samplings that are used in circuit simulations.* In the latter, the input space is sampled, whereas in our case both the input and the line state spaces are sampled simultaneously, using a strong correlative model, as captured by the Bayesian Network. Due to this, convergence is faster and the inference strategy is input pattern insensitive.

4.3 Experimental Results

We experimented with the combinational circuits from the ISCAS85 benchmark suite. We first mapped the ISCAS circuits to their corresponding DAG structured Bayesian Networks. Each node in the Bayesian Network takes four possible outcomes (x_{00}, x_{01}, x_{10}, x_{11}). The conditional probability of each node is formed based on the knowledge of type of gate connecting the parent and the child. The experimental set-up of ”GeNIe” [22], a graphical network interface is used for our experimentation. The tests were performed on a Pentium IV, 2.00GHz, Windows XP computer. For comparison, we performed zero-delay logic simulation on the ISCAS85 benchmark circuits, which provides accurate estimates of switching.

Table 4.1. shows the mean, standard deviation, maximum error and the time elapsed for the ISCAS circuits with PLS and we compare the results with that obtained using the prior approximate Cascaded Bayesian Networks (CBN) [5]. Columns 2, 3, 4 and 5 in this table represents the mean error (μ_E), standard deviation of the error (σ_E), maximum error (Mx_E), and the elapsed time (T) for switching activity. It can be easily seen that even though good mean errors are obtained by approximate CBN methods, the stochastic PLS provides better estimates in terms of standard deviation and shows significant improvement over the maximum error. The total elapsed time, which is the sum of CPU, memory access and I/O time (computed using the ftime command in WINDOWS environment) is also significantly low for PLS. High maximum error in the approximate Cascaded Bayesian Network (CBN) model is attributed to partitioning of the network, which results in loss of information at the boundary nodes.

The PLS scheme converges to accurate estimates when propagating evidence along the causal links, but for diagnostic reasoning the estimates obtained through this approach deteriorates. Tables 4.2. and 4.3. show the error statistics for predictive as well as diagnostic inference using Adaptive Importance Sampling and Evidence Pre-propagation Importance Sampling for 500, 1000 samples. Comparison of both the tables show the two algorithms converge close to accurate estimates within 500 samples. The mean and standard deviation of the error and the maximum error are extremely low for both the models even for larger benchmark circuits like c3540, c6288. This can be attributed to the formation of a good importance function that is close to the optimal importance function. However, EPIS shows faster convergence than AIS as it avoids the costly learning process in AIS algorithm. Note that the diagnostic feature that both AIS and EPIS methods offer over the Approximate Cascaded Bayesian Network methods is one of our key motivations for using stochastic inference.

Figures 4.2., 4.3., and 4.4., corresponding to c432, c1355, c6288 benchmark circuits, respectively, show the variation of errors, obtained using AIS, EPIS and PLS. Analysis of the graph shows that the estimates converge faster within a small sample space and estimates can always be formed even when the sample space is small or insufficient (any-time).

c432

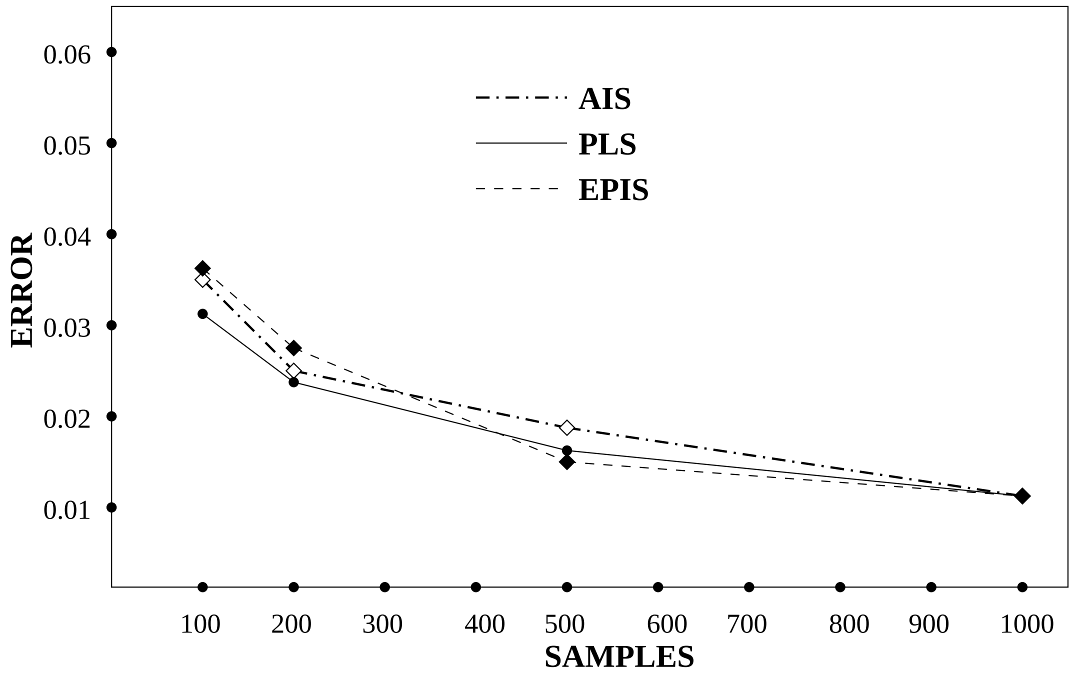


Figure 4.2. Graph showing the time accuracy trade off for c432.

c1355

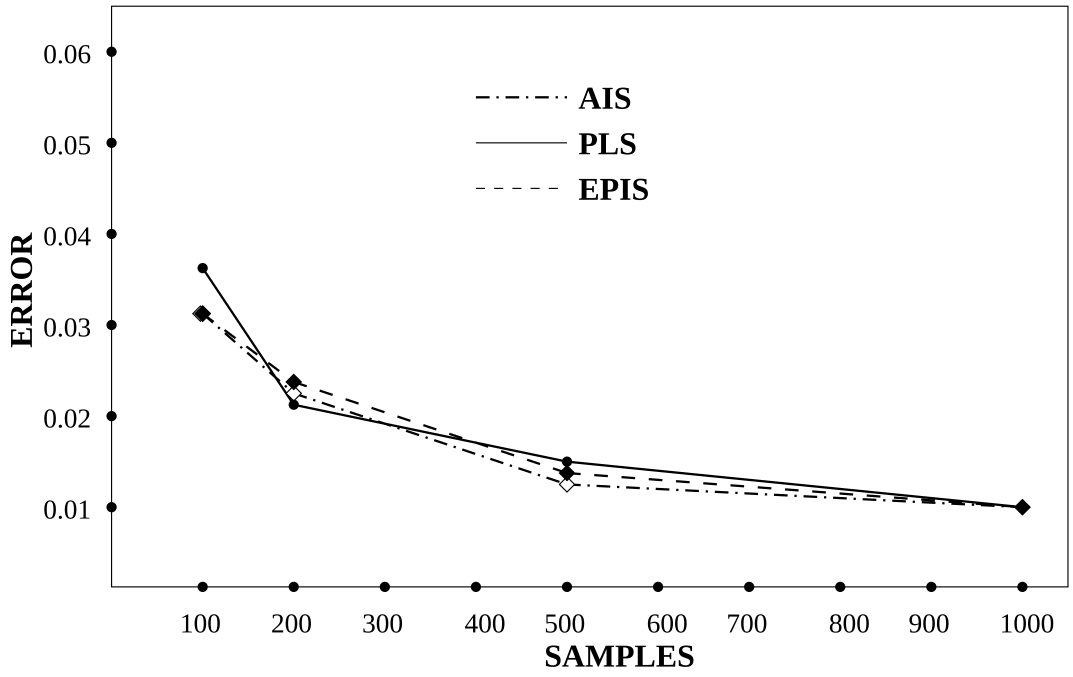


Figure 4.3. Graph showing the time accuracy trade off for c1355.

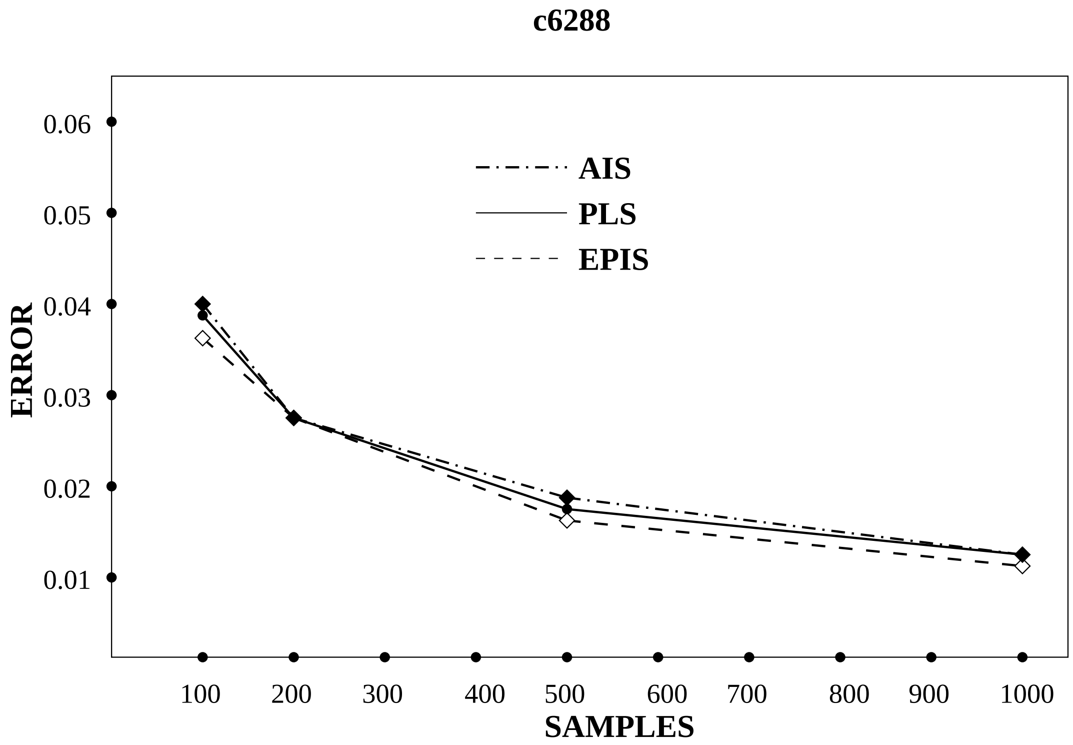


Figure 4.4. Graph showing the time accuracy trade off for c6288.

Table 4.1. Experimental results comparing Approximate Cascaded Bayesian Network model and Probabilistic Logic Sampling.

	Approx. CBN model [5]				PLS: 1000 samples			
	μ_E	σ_E	Mx_E	T(s)	μ_E	σ_E	Mx_E	T(s)
c432	0.00	0.02	0.28	3	0.00	0.00	0.04	0.40
c499	0.00	0.00	0.00	9.03	0.00	0.01	0.04	0.45
c880	0.00	0.00	0.04	2.52	0.00	0.01	0.05	1.05
c1355	0.00	0.00	0.09	1.81	0.00	0.01	0.06	1.75
c1908	0.00	0.01	0.15	10.70	0.00	0.01	0.05	2.7
c3540	0.00	0.04	0.26	18.86	0.00	0.00	0.04	5.96
c6288	0.01	0.04	0.37	38.75	0.00	0.01	0.06	11

Table 4.2. Experimental results using AIS algorithm for various samples.

	AIS: 500 samples				AIS: 1000 samples			
	μ_E	σ_E	Mx_E	T(s)	μ_E	σ_E	Mx_E	T(s)
c432	0.004	0.014	0.056	16.42	0.001	0.009	0.041	16.68
c499	0.001	0.012	0.097	19.42	0.000	0.009	0.041	19.67
c880	0.000	0.013	0.057	40.29	0.000	0.010	0.043	40.82
c1355	0.001	0.013	0.064	62.48	0.000	0.009	0.052	63.68
c1908	0.002	0.015	0.069	97.75	0.000	0.010	0.044	99.62
c3540	0.001	0.012	0.065	205.7	0.001	0.009	0.048	212.8
c6288	0.001	0.014	0.085	389.33	0.002	0.010	0.056	394.78

Table 4.3. Experimental results using EPIS algorithm for various samples.

	EPIS: 500 samples				EPIS: 1000 samples			
	μ_E	σ_E	Mx_E	T(s)	μ_E	σ_E	Mx_E	T(s)
c432	0.004	0.011	0.049	0.72	0.002	0.009	0.048	1.04
c499	0.001	0.012	0.055	0.94	0.001	0.008	0.039	1.03
c880	0.000	0.014	0.078	2.82	0.002	0.010	0.056	3.36
c1355	0.002	0.019	0.056	6.95	0.001	0.009	0.051	7.82
c1908	0.004	0.015	0.067	15.42	0.001	0.009	0.044	16.64
c3540	0.002	0.013	0.070	52.34	0.001	0.009	0.042	54.76
c6288	0.002	0.012	0.069	143.23	0.001	0.009	0.052	144.33

CHAPTER 5

ENTROPY BASED INPUT CHARACTERIZATION FOR DATA-DEPENDENT LEAKAGE POWER ANALYSIS

The average power dissipation in CMOS is sum of two factors, namely, dynamic power and leakage power. Tradiationally, the most significant contributor to power dissipation in CMOS circuits has been the dynamic power dissipation. To attenuate this problem designers relied on scaling down the supply voltage due to the quadratic dependence of dynamic power on supply voltage. A major disadvantage of this technique is that it affects the switching speed of the circuit. Sustenance of the circuit speed requires a proportionate decrease in the threshold voltage. Downscaling of the threshold voltage aggravates leakage power dissipation (due to an exponential increase in sub-threshold leakage current). Also, as the device density increases leakage power starts dominating the total power dissipation. Leakage power dissipation can account for more than 50% of total power dissipation in 65nm IC's. An effective strategy for mitigating leakage power is to use dual threshold voltage cells (placing low threshold voltage cells in the critical path to maintain performance and having high threshold voltage cells in the non-critical paths to reduce leakage). It shoulf be noted that, leakage does not only happen during stand-by modes, it is also present during active period and is a crucial component of total power dissipation [61], [62], [63]. This chapter introduces a novel technique using Bayesian Networks that identifies gates which are dormant even in the run-time mode. Designers can target these gates as candidate gates for leakage power optimization. The intension of this chapter is not to focus on the methods of leakage optimization but to provide designers with another technique to target gates for leakage power optimization.

5.1 Why do we need Input Characterization?

The total power (P_t) expended in a circuit can be expressed as the sum of individual gate power (P_{tg}), which in turn can be broken up into switching and leakage components [61].

$$\begin{aligned} P_t &= \sum_g P_{tg} = P_{dg} + P_{sg} \\ &= 0.5\alpha f V_{dd}^2 C_{load+wire} + (1-\alpha)\sum_i P_{leak,i}\beta_i \end{aligned} \tag{5.1}$$

where α denotes the activity of the node, and β is the probability of remaining in a dominant leakage state (namely signal at 0 or 1). Note that β as well as α is dependent on $\{y_i\} \in \{0_{t-1}0_t, 0_{t-1}1_t, 1_{t-1}0_t, 1_{t-1}1_t\}$ the switching states of the inputs and the physical parameters of the gate, θ. Thus, total power is a function of the input switching states, $(y_1, \cdots, y_N)$, and the device parameters, θ, i.e. $P_t(y_1, \cdots, y_N, \theta)$. Given an input trace, one usually computes $\sum_{(y_1,\cdots,y_N)} P_t(y_1, \cdots, y_N, \theta)$, which is akin to computing the expected value of the total power, $E(P_t)$, with the particular input trace. The optimization problem can then be expressed as:

$$\min_\theta E(P_t) = \min_\theta \sum_{y_1,\cdots,y_N} P_t(y_1, \cdots, y_N, \theta) p(y_1), \cdots, p(y_N) \tag{5.2}$$

The optimal point, θ_{opt}, thus found will be a function of the assumed input statistics, $p(y_1), \cdots, p(y_N)$ [1]. Note that, power optimized operating point arrived at for a single input switching point, or a few set of input statistics, is not sufficient. The space of all possible input switching statistics is just too large, in fact it is much larger than the size of the input space itself; it is the set of all possible probability distributions over not the input states, but the input switching states. Moreover, simplistic random input assumptions on the input space is a serious problem. Inputs of one circuits are the outputs of another, and hence can exhibit strong correlations. For instance, Figure. 5.1. shows the probabilistic dependencies amongst ten of the outputs of c432 (4 bit ALU) benchmark. We provide random inputs to the primary inputs of c432 and learn a causal probabilistic graphical model amongst the outputs (the learning process used is not in the scope of this thesis). The probabilistic dependency between these nodes definitely are far from random and even modeling using just biased inputs (low/high switching) is not sufficient. Clearly, there is need for a statistical input characterization measure for leakage

[1] In the chapter, we use capitals, e.g. Y, to denote random variables, corresponding small letters, e.g. y to denote values. We also use $p(y)$ to denote $p(Y = y)$, i.e. the probability of the event $Y = y$

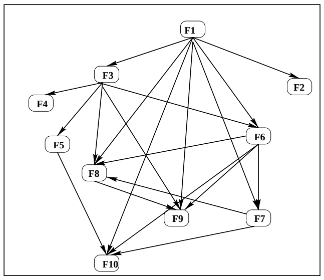


Figure 5.1. The correlations among the output lines of c432 with random inputs. The output lines of one block are the input lines of another.

optimization that is not based on prior knowledge of input trace. Of course, if complete data trace specifications for an application domain are known, any estimation algorithm can predict the likelihood of a signal to leak. However, in practice this is hard to accomplish, especially in the nano-domain, where due to the multi-objective optimization needs, the size of the data trace requirements increase, so as to be able to exercise all the "modes" of the objective functions.

Instead of considering total power based on a *prior* over the input space, which has been the usual practice, we consider the *posterior* over the input space, conditioned on states of internal nodes. Using these posterior distributions over the inputs, we identify nodes that are likely to be leaky in the active zone. It does *not* require the pre-specification of input-statistics or input-traces. Mathematically, we consider $p(\{y_i\}|X_j = 0_{t-1}0_t)$, where X_j represents a random variable internal to the circuit and $p(\{y_i\}$ is the profile of input set that generates the observation $X_j = 0_{t-1}0_t$. We use the concept of entropy of this posterior distribution to characterize the input space. The concept of entropy in itself is not new and is used in almost all research disciplines. Excellent introduction to axiomatic basis of entropy concepts can be found in [73, 74, 75]. Entropy is a measure of uncertainty in a finite system. High entropy indicates high uncertainty. Completely random inputs would be associated with high entropy. In this work, we generate an upper bound of the entropy of the posterior input space distribution by considering independent inputs. if no prior information about the input is known, the concept of absolute entropy can be used. However, if some knowledge of the input is available, such as when one knows that the inputs are the outputs of another logic block, then in such case, we can use relative entropy as a distance measure between the known input space and the posterior input space under the evidence/observation.

We use Bayesian Belief Network based modeling to compute the input posteriors. Why Bayesian Networks? Simply because these causal, graphical models are *minimal* representations completely capturing the underlying joint probability density function (pdf); it induces an optimal factorization of the joint pdf [2]. The minimality of the representation manifests in the reduced number of the links in the graph representation, which in turn facilitates fast belief propagation schemes. These probabilistic belief propagation schemes are not direction-sensitive, even though the underlying graph representation has directed links between cause and effect; during updating messages are passed in both directions. In fact, this probabilistic model is pattern-insensitive for predictive purposes, where we know the possible causes and we want to know the effects. But, the real power of probabilistic models is that given an observation, we can *characterize* the plausible causes that can produce the given observation, that is, the probabilistic models provide the *Most Probable Explanation* that can cause the observation. We use Evidence Pre-propagated Importance Sampling [12] stochastic inference technique to accurately propagate belief from evidence to all other variables.

The issue and the approach presented in this chapter are new to our knowledge. For a different context of dynamic power, data specification, or rather lack of specification, issues has been studied using power sensitivity [76], where upper/lower bounds on average dynamic power was computed. Note that this type of treatment depends on a few simulation-based (pattern-sensitive) points, around which sensitivity is computed and treats only the dynamic component of power. With regard to the concept of entropy, it has been used in RT power as an upper bound of switching activity by Nemani *et al.* [77] and not for input space characterization. With regards to Bayesian networks, it was first proposed in [8] and then in [78]. However, the contributions have been limited to switching estimation and timing prediction that do not exploit the backtracking aspect of probabilistic reasoning. Our work provides another arsenal for the low power designers to focus leakage mitigation schemes at targeted nodes, rather than just considering nodes (based on critical path) or using a single (or a handful of data profiles) to target the optimization schemes. Methods that use measures on every transistors, such as dynamic threshold voltage [67, 68], can use our measure to select the target nodes.

5.2 Input Characterization

We approach the input space characterization problem as a process to find out the plausible causes of an observation. In our case, suppose we want to get a measure of leakiness ($X_i = 0_{t-1}0_t, X_i = 1_{t-1}1_t$) or amount of switching ($X_i = 0_{t-1}1_t, X_i = 1_{t-1}0_t$) of an internal signal X_i, predictive inference strategies would actually consider the conditional probability $p(X_i = 0_{t-1}0_t | y_1, \cdots, y_N)$ where knowledge of input switching states $y_1, \cdots, y_N$) are assumed to be known *apriori*. On the other hand, diagnostic inference would analyze $p((y_1, \cdots, y_N)|X_i = 0_{t-1}0_t)$ and then use the posterior input distribution to characterize the input space for the observation $(X_i = 0_{t-1}0_t)$. Before we discuss, in Section 5.3, how to compute this input posterior, we present the entropy based characterization that we advocate. Note that, we will be using uppercase alphabets (namely X, Y) to denote random variables and lowercase alphabets (namely x, y, y_i, y_j) to denote the discrete states that the random variable can take.

Before providing the definitions, let us intuitively see why we use entropy. Entropy of a system measures the amount of uncertainty in the system. A more likely event is bound to have higher entropy than a less likely event [74]. Consider a random variable Y_i, representing the i-th input. For our case, each Y_i can taken on one of four possible switching states values, $y_i \in \{0_{t-1}0_t, 0_{t-1}1_t, 1_{t-1}0_t, 1_{t-1}1_t\}$. Entropy of this random variable is given by

$$H(Y_i) = -\sum_{y_i} p(y_i) \log p(y_i) \tag{5.3}$$

If random switching, $p(y_i) = 1/4$ and $H(Y_i)$ is $\log 4$, the maximum possible. If the state of variable is known for sure, $H(Y_i)$ is zero.

Entropy can also be computed for a set a random variable by considering their joint probability function and it can be shown that [79].

$$H(Y_1, \cdots, Y_N) = \sum_{i=1}^{N} H(Y_i | (Y_1, \cdots, Y_{i-1}, Y_{i+1}, \cdots, Y_N)) \tag{5.4}$$

When the random variables are independent of each other joint entropy is such the sum of the individual entropies

$$H^u(Y_1, \cdots, Y_N) = \sum_{i=1}^{N} H(Y_i) \tag{5.5}$$

It can be shown that $H^u \geq H$ where y_i's are not mutually independent, or is *an upper bound on the joint entropy*. We use this upper bound entropy measure on the posterior input space as a leakage possibility characterization measure for node, X_j.

$$H^u_{x_j}(Y_1, \cdots, Y_N) = -\sum_{i=1}^{N} p(y_i|x_j) \log p(y_i|x_j) \tag{5.6}$$

We can use entropic measures, even for situations where we might know something about the input statistics. In such cases we consider the relative entropy, which is also known as the Kullback-Liebler distance, cross entropy, discrimination information, directed divergence, or I-divergence. Let $q(y_1), \cdots, q(y_N)$, denote the known input statistics. Then the relative entropy between the posterior distribution and the known one is given by

$$H^r_{x_j} = \sum p(y_1, \cdots, y_N|x_j) \log \frac{p(y_1, \cdots, y_N|x_j)}{q(y_1), \cdots, q(y_N)} \tag{5.7}$$

This serves as a measure of leakage possibility. If the relative entropy is high between the posterior input space, conditioned on a leaky state, x_j, of a node, X_j, and the given input space, then it is unlikely that particular node, X_j, will leak. And vice versa, if the distance is less.

Like for simple entropy, it can be shown that the relative entropy measure under independence assumption can provide us with an upper bound. The upper bound relative entropy can expressed simply as

$$\begin{aligned}
H^r_{x_j} &= \sum p(y_1, y_2, \cdots, y_N) \log \frac{p(y_1, y_2, \cdots, y_N)}{q(y_1, y_2, \cdots, y_N)} && (5.8)\\
&= \sum p(y_1)p(y_2)\cdots p(y_N) \sum_i \log \frac{p(y_i)}{q(y_i)} && (5.9)\\
&= \sum_{i,s} p(y_i) \log \frac{p(y_i)}{q(y_i)} \prod_{\forall j \neq i} (\sum_s p(y_j)) && (5.10)\\
&= \sum_{i,s} p(y_i) \log \frac{p(y_i)}{q(y_i)} && (5.11)
\end{aligned}$$

Equation. 5.9 exploits the factorization that is possible based on independence. In Eq. 5.10, $\sum_s p(y_j)) = 1$. Note that under the independence assumption, the relative entropy computation is inexpensive.

5.3 Computing Posterior Input Distributions

In this section, we discuss the probabilistic model and the backward propagation schemes that are used to compute the posterior distributions over the input space. Any probability function over a set of random variables $(X_1, \cdots, X_M, y_1, \cdots, y_N)$, where X_i, are the random variables representing i-th internal signals, Y_j is the j-th primary input, can be represented as

$$\begin{aligned}&P(x_1, \cdots, x_M, y_1, \cdots, y_N)\\&= P(x_M|x_{M-1}, x_{M-2}, \cdots, x_1, y_1, \cdots, y_N)\\&P(x_{M-1}|x_{M-2}, x_{M-3}, \cdots, x_1, y_1, \cdots, y_N), \cdots, P(y_1). \cdots P(y_N))\end{aligned} \tag{5.12}$$

The above expression holds for any ordering of the random variables. This brute force representation of probabilistic knowledge requires tabulation of all the entries in $P(x_1, \cdots, x_M, y_1, \cdots, y_N)$, which are exponential in number. The exact representation assumes that every variable is dependent on every other random variable present in the set. It does not take advantage of the conditional independencies present among the variables. Until very recently, the common strategy has been to make the naive assumption of complete independence of the variables, i.e. $P(x_1, \cdots, x_M) = P(x_1), \cdots, P(x_M)$. However, this fails to exploit the full power of probabilistic modeling. Powerful directed acyclic graph (DAG) based models have been recently proposed that exploits *all* the independencies among the random variables to arrive at a *minimal* model of the joint probability graph. These are termed Bayesian networks, causal networks, or belief networks [2]. This graphical probabilistic model induces a factored representation of the joint probability function in term of the conditional probabilities of each random variables, given the states of their corresponding parents in the graph structure.

$$P(x_1, \cdots, x_M, y_1, \cdots, y_N) = \prod_{k=1}^{M} P(x_k|Pa(x_k)) \prod_{j=1}^{N} P(y_j) \tag{5.13}$$

The power of Bayesian Networks is in the fact that not only can it do predictive inference, i.e. compute probabilities of the outputs (effects) based on evidence about the inputs (causes), but it can also do diagnostic inference, i.e. compute probabilities about the inputs (causes) conditioned on the outputs (effects). While simulative approaches, such as those used in VLSI estimation, might suffice for predictive inference, they are not at all efficient for diagnostic inference. One example of diagnostic query is “What inputs can cause a switching probability of 0.8 at a particular node of interest?” This would require a *conditional* search of the input space in a simulative framework, which is computationally expensive. Probabilistic inference is the only consistent uncertainty calculus to handle such situations in an efficient manner. We use EPIS (a stochastic sampling technique discussed in chapter 4, section 4.2.3) technique to estimate the posterior input distributions given an observation. For our experiments, we used the Bayesian network computational package and library known as ”GeNIe” [22] and SMILE.

Note that this sampling based probabilistic inference is non-simulative and is different from samplings that are used in circuit simulations. In the latter, the input space is sampled, whereas in our case both the input and the line state spaces are sampled simultaneously, using a strong correlative model, as captured by the Bayesian Network. Due to this, convergence is faster and the inference strategy is input pattern insensitive.

5.4 Results and Conclusions

We illustrate the ideas presented in this chapter using the ISCAS’85 benchmark circuits. All computations are run on a Pentium IV, 2.00GHz, Windows XP computer. Table 5.1. shows the profiles of nodes that are tested for known data traces. We tested with two sets. The first set is where the input switching is moderate (switching activity 0.5) and the second set is for a biased low switching inputs (switching activity 0.3). Column 2 and 3 denotes the number of nodes that would be leaking 60-80% and above 80% of the time during the active mode of operation (input switching is assumed to 0.5). For an example c3540, would have 841 nodes leaking more than 80% time and c1908 has 362 nodes leaking for more than 80 % of time. Note that the above data is reported on individual signals. However, for an NMOS stack, one of the input at 1 might not make it a leaky state. For the true computation, we should consider the joint instantiation of the inputs to either $0_{t-1}0_t$ or to $1_{t-1}1_t$ and then calculate entropy. Note that the calculation effort of such a combination would not be different from single instantiation.

Figure. 5.2., 5.3., 5.4. shows the detailed profile of the node breakup of the percentage of time leaking at state zero, for three circuits and with two different input statistics; for one case, primary inputs are switching 50% of the time and in the other case, primary inputs are switching 30% of time.

Table 5.1. Leakage profile for known data traces.

Circuits	Number of nodes leaking in active zone			
	Input activity 0.5		Input activity 0.3	
	(60-80)% of time	$\geq$ 80% of time	(60-80)% of time	$\geq$ 80% of time
c432	178	37	363	102
c499	54	88	235	88
c880	252	123	610	224
c1355	402	280	743	306
c1908	372	362	1224	395
c3540	896	841	2068	1040
c6288	1997	292	2856	555

Notice that a larger number of nodes would be leaking significantly during active zone for activity 0.3. Data shown in Table 5.1., as well as in Figure. 5.2., 5.3., 5.4., are generated by the same probabilistic framework and inference that is used for the posterior computation. All the data are non-simulative, hence pattern insensitive and models *all dependencies* in the circuits.

In Table 5.2., we show the results on input space characterization by absolute entropy. Note that this table is for the entropy computation when posterior input distribution were computed forcing individual signals to leak at logic zero. Signals that produced an entropy measure that lies within 30% of the maximum entropy are reported at column 3 and these nodes are targets for leakage mitigation schemes. Minimum and maximum entropy is reported in column 4 and 5 respectively. Column 2 reports nodes are highly unlikely to leak at zero: these signals produced entropy that within 30% of the lowest entropy seen for those circuits. Note that for c880 and c1908, there are nodes that do not leak at any time at zero for any input combinations. Some circuits showed larger range in entropy than others. For example, in c3540 the difference between maximum and minimum entropy is 10 whereas c499 and c1355 entropy range is small. This indicates the data traces are more important to obtain for the latter groups (c499). Also, we found out a few nodes that cannot be leaking at zero irrespective of the input space. This happens when posterior distribution is impossible.

In Table 5.3., we present results on the relative entropy. Note that, relative entropy measures distance from a reference distribution and hence signals that generates low H^r, are more likely to be leaky at zero given the expected reference distribution. We report number of nodes that generated high input KL distance measure (within 30% of the maximum H^r) in column 2 and number of nodes that

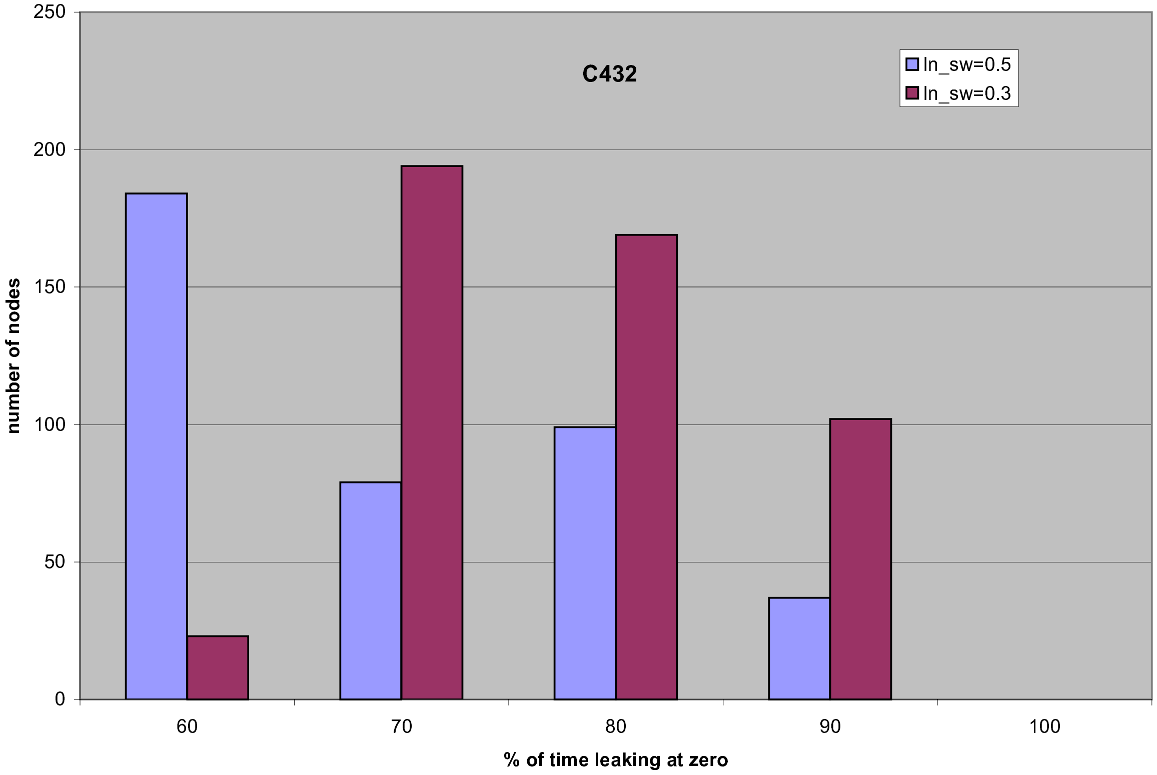


Figure 5.2. Graph showing the break-up of leakage profile for different switching profile for c432.

Table 5.2. Entropy based input characterization for leakage condition

Circuits	Number of nodes generating		Range of Entropy	
	Low input entropy	High input entropy	min	max
c17	4	2	4.14	6.92
c432	2	321	44.66	49.86
c499	31	321	53.95	56.78
c1355	128	793	53.95	56.79
c1908	4	1871	0	45.74
c3540	5	3220	58.91	69.25
c6288	6	4172	37.81	44.33

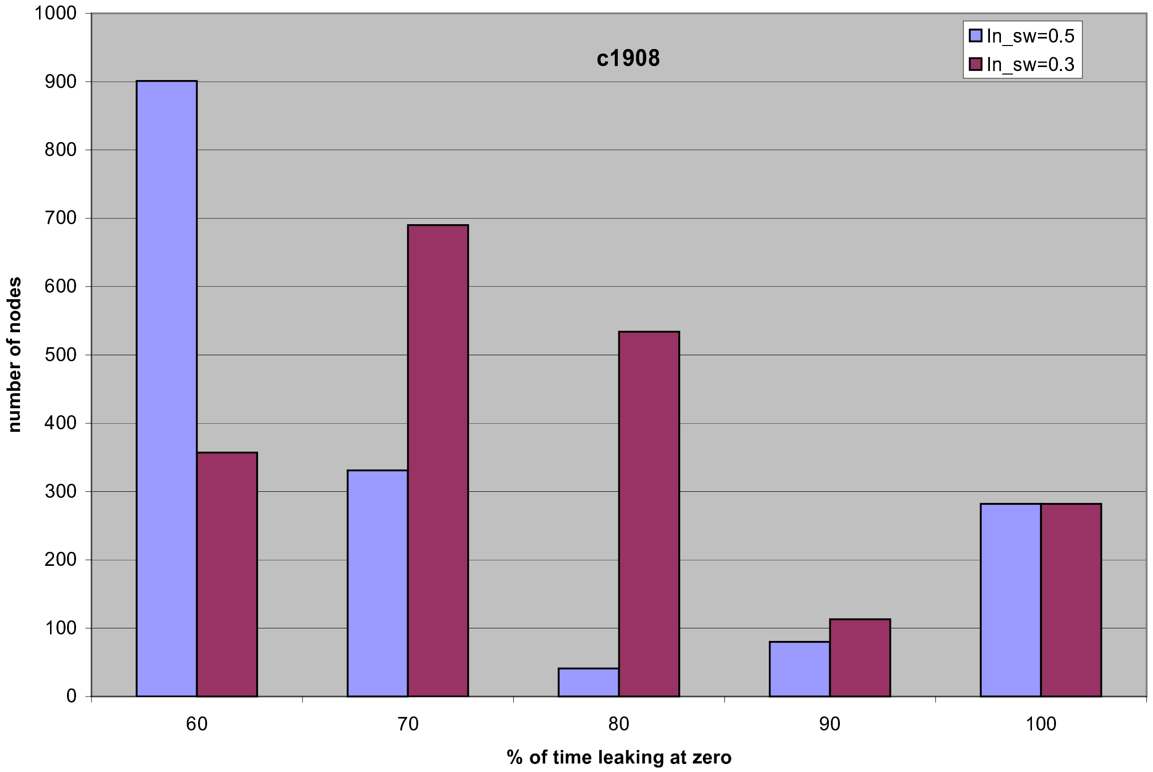


Figure 5.3. Graph showing the break-up of leakage profile for different switching profile for 1908.

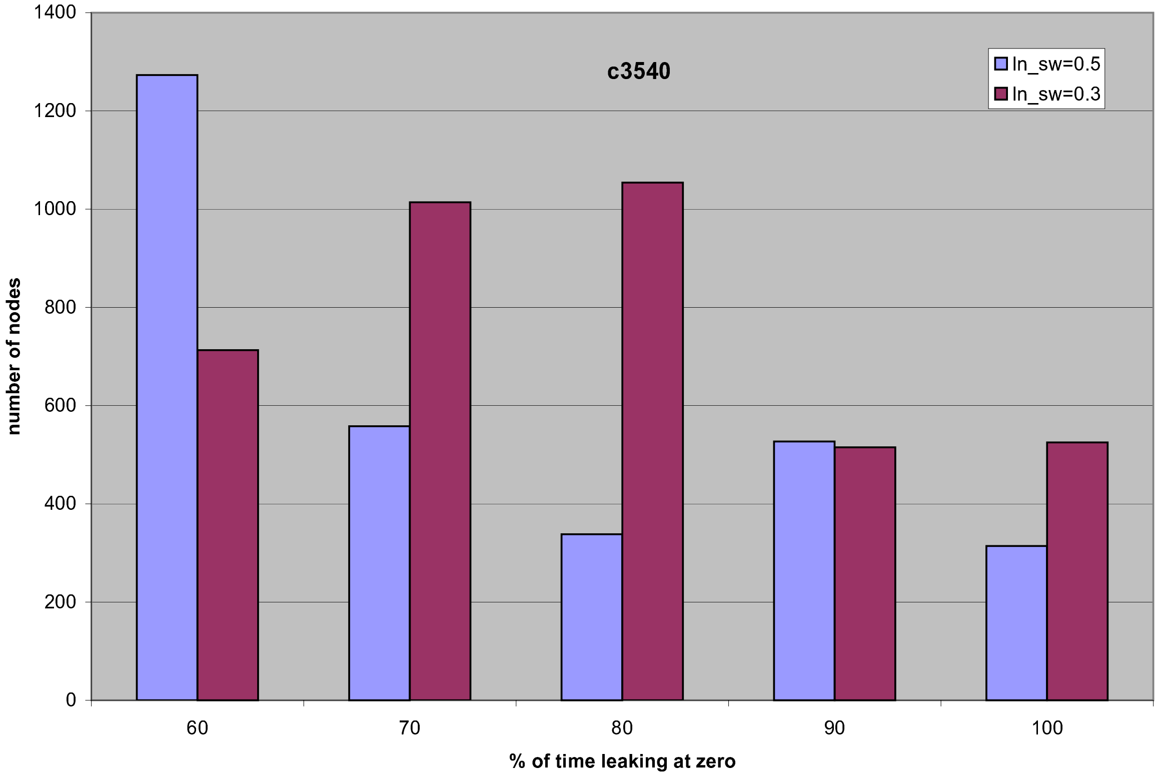


Figure 5.4. Graph showing the break-up of leakage profile for different switching profile for c3540.

Table 5.3. Relative entropy based input characterization for leaky signals

Circuits	Number of nodes generating		Range of H^r	
	High H^r	Low H^r	Min H^r	Max H^r
c17	4	2	0.01	2.78
c432	1	303	0.04	4.77
c499	30	322	0.05	2.88
c1355	127	794	0.05	2.88
c1908	13	1760	0.04	5.6
c3540	4	3221	0.06	10.39
c6288	5	4175	0.04	6.55

generated low (bottom 30%) input H^r measure in column 3. Columns 4 and 5 reports, minimum and maximum relative entropy, respectively, that we observe for the circuits.

CHAPTER 6

CONCLUSION

Switching activity estimation is a complex problem that have been researched for more than a decade. This thesis introduces a switching activity estimation tool for combinational circuits that models all the temporal and spatial dependencies in a circuit with high accuracy. We have demonstrated the results of the estimated switching activity using various any-time stochastic sampling inference algorithms namely EPIS, AIS, and PLS. We find that PLS yields the best accuracy-time tradeoff if used under predictive situation. In diagnostic situation, cases when evidence is unlikely, EPIS and AIS algorithms would yield accurate estimates. We thus conclude that the Bayesian Network based modeling of switching activity and inference yields higher accuracy in significantly lower time. The present scope of this model is limited to zero-delay scenario, which we plan to address in future.

We also presented a novel probabilistic framework for measuring leakiness potential of a signal during active switching mode, without any prior assumption about the input space. We use the probabilistic Bayesian Network to propagate belief from observation to the plausible causes and use the attributes like entropy of the posterior to determine the likeliness of the signals to be at zero. Our work provides another arsenal for the low power designers to focus leakage mitigation schemes at targeted nodes, rather than just considering nodes (based on critical path) or using a single (or a handful of data profiles to target optimization scheme. Methods, such as dynamic threshold voltage, can use our measure to select the target nodes for leakage optimization.

REFERENCES

[1] URL `http://www.intel.com`

[2] J. Pearl, ”Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference”, Morgan Kaufmann Publishers, Inc., 1988.

[3] Gary K. Yeap, ”Practical Low Power Digital VLSI Design”, Kluwer Academic Publishers, Inc., 1998.

[4] H. J. Veendrick, ”Short-Circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of Buffer Circuits”, *Journal of Solid-State Circuits*, vol. SC-19, no. 4, pp. 468-473, August 1984.

[5] S. Bhanja and N. Ranganathan, ”Modelling Switching Activity using Cascaded Bayesian Networks for correlated Input streams”, *International Conference on Computer Design*, 2002.

[6] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, ”On average power dissiation and random pattern testability of CMOS combinational logic networks”, *IEEE/ACM International Conference on Computer-Aided Design*, Santa Clara, CA, November 8-12, 1992, pp. 402-407.

[7] Semiconductor Industry Association, International Roadmap for Semiconductors, 2002 Update; http://public.irts.net.

[8] S. Bhanja and N. Ranganathan, “Dependency Preserving Probabilistic Modeling of Switching Activity using Bayesian Networks”, *38th IEEE/ACM Design Automation Conference*, pp 209-214, 2001.

[9] Shiva Shankar Ramani, and S. Bhanja, ”Any-time Probabilistic Switching Model using Bayesian Networks”, *International Symposium on Low Power Electronics and Design*, pp. 86-89, Newport Beach, CA, 2004.

[10] M. Henrion, ”Propagating Uncertainity in Bayesian Networks by Probabilistic Logic Sampling”, In *Uncertainity in Artificial Intelligence 2*, pp. 149-163, New York, N.Y., 1988. Elsevier Science Publishing Company, Inc.

[11] J. Cheng, ”Efficient Stochastic Sampling Algorithms for Bayesian Networks”, *Ph.D Dissertation, University of Pittsburgh*, 2001.

[12] C. Yuan, and M. J. Druzdzel, ”An Importance Sampling Algorithm Based on Evidence Prepropagation”, *Proceedings of the 19th Annual Conference on Uncertainty on Artificial Intelligence*, pp. 624-631, 2003.

[13] R. G. Cowell, A. P. David, S. L. Lauritzen, and D. J. Spiegelhalter, “Probabilistic Networks and Expert Systems”, Springer-Verlag New York, Inc., 1999.

[14] Kaushik Roy, Saibal Mukhopadhyay, and Hamid Mahmoodi-Meimand, ”Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits”, Proceedings of the IEEE, Vol. 91, No. 2, February 2003.

[15] Y. Taur and T. H. Ning, ”Fundamentals of Modern VLSI devices”, New York: Cambridge Univ. Press, 1998.

[16] G.F. Cooper, ”The computational complexity of probabilistic inference using Bayesian belief networks”, *Artificial Intelligence*, 42(2-3):393-405, March 1990.

[17] P. Dagum and M. Luby, ”Approximating probabilistic inference in Bayesian belief networks is NP-hard”, *Artificial Intelligence*, 60(1):141-153, 1993.

[18] R. Fung and K.-C. Chang, ”Weighting and integrating evidence for stochastic simulation in Bayesian networks”, In M. Henrion, R.Shachter, L. Kanal, and J. Lemmer, editors, *Uncertainity in Artificial Intelligence 5*, pages 209-219, New York, N. Y., 1989. Elsevier Science Publishing Company, Inc.

[19] R. D. Shachter and M. A. Poet, ”Simulation approaches to general probabilistic inference on belief networks”, In M. Henrion, R.Shachter, L. Kanal, and J. Lemmer, editors, *Uncertainity in Artificial Intelligence 5*, pages 221-231, New York, N. Y., 1989. Elsevier Science Publishing Company, Inc.

[20] R. Fung and B. del Favero, ”Backward simulation in Bayesian networks”, In *Proceedings of the Tenth Annual Conference on Uncertainity in Artificial Intelligence (UAI-94)*, pages 227-234, San Mateo, CA, 1994. Morgan Kaufmann Publishers, Inc.

[21] K. Murphy, Y. Weiss, and M. Jordan, ”Loopy Belief Propagation for Approximate Inference: An Empirical study”, *Proceedings of the Conference on Uncertainity in AI(UAI-99)*, pp. 467-475, 1999.

[22] "Graphical Network Interface",
URL http://www.sis.pitt.edu/˜genie/genie2

[23] URL http://www.hugin.com/

[24] S. Bhanja, and N. Ranganathan,”Switching Activity Estimation of VLSI Circuits using Bayesian Networks”, Accepted in *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, 2002.

[25] R. Burch, F. N. Najm, and T. Trick, ”A Monte Carlo Approach for Power Estimation”, *IEEE Transactions on VLSI Systems*, vol. 1-1, pp. 63-71, March 1993.

[26] F. N. Najm, “Transition Density: A New Measure of Activity in Digital Circuits”, *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 2, pp. 310–323, February 1993.

[27] F. N. Najm, R. Burch, P. Yang, and I. N. Hajj, “ Probabilistic Simulation for Reliability Analysis of CMOS Circuits”, *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 4, pp. 439–450, April 1990.

[28] R. E. Bryant, ”Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams”, *ACM Computing Surveys*, vol. 24, No. 3, pp. 293-318, Sept. 1992.

[29] R. Marculescu, D. Marculescu, and M. Pedram, ”Probabilistic Modeling of Dependencies During Switching Activity Analysis”, *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17-2, pp. 73-83, February 1998.

[30] A. Ghosh, S. Devadas, K. Keutzer, and J. White, ”Estimation of Average Switching Activity in Combinational and Sequential Circuits”, *Proceedings of the 29th Design Automation Conference*, pp. 253–259, June 1992.

[31] M. A. Cirit, “Estimating Dynamic Power Consumption of CMOS Circuits”,*IEEE International Conference on Computer -Aided Design*, pp. 534–537, 1987.

[32] K. S. Brace, R. L. Rudell, and R. E. Bryant,” Efficient implementation of a BDD package”, *27th ACM/IEEE Design Automation Conference*, pp. 40–45, 1990.

[33] L. Benini, M. Favalli, P. Olivo, and B. Ricco, “A novel approach to cost-effective estimate of power dissipation in CMOS ICs,” *European Design Automation Conference*, pp. 354–360, 1993.

[34] S. Chakravarti, “ On the Complexity of using BDDs for the Synthesis and Analysis of Boolean Circuits”, *Proceedings of 27th Annual Conference on Communication, Control and Computing*, pp. 730–739, 1989.

[35] C.-S. Ding, C.-Y. Tsui, and M. Pedram, “Gate-Level Power Estimation Using Tagged Probabilistic Simulation”, *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 11, pp. 1099–1107, November 1998.

[36] K. Parker, and E. J. McCluskey, “Probabilistic Treatment of General Combinational Networks”, *IEEE Trans. on Computers*, vol. C, no. 24, pp. 668–670, June 1975.

[37] G. I. Stamoulis, and I.N. Hajj, “ Improved Techniques for Probabilistic Simulation including Signal Correlation Effects”, *Proceedings of the 30th ACM/IEEE Design Automation Conference*, pp. 379–383, June, 1993.

[38] S. M. Kang, “Accurate Simulation of Power Dissipation in VLSI Circuits”, *IEEE Journal of Solid-state Circuits*, vol. SC-21, no. 5, pp. 889–891, Oct. 1986.

[39] T. H. Krodel, “PowerPlay fast dynamic power estimation based on logic simulation,” *IEEE International Conference on Computer Design*, pp. 96–100, October 1991.

[40] A. K. Murugavel, and N. Ranganathan, “Petri Net Modeling of Gate and Interconnect Delays for Power Estimation”, *Proc. of the Design Automation Conference*, pp. 455–460, 2002.

[41] A. C. Deng, Y. C. Shiau, and K. H. Loh, “Time Domain Current Waveform Simulation of CMOS Circuits”, *IEEE International Conference on Computer Aided Design*, Santa Clara, CA, pp. 208–211, Nov. 7-10, 1988.

[42] F. Rouatbi, B. Haroun, and A. J. Al-Khalili,” Power estimation tool for sub-micron CMOS VLSI circuits”, *IEEE/ACM International Conference on Computer-Aided Design*, pp. 204–209, 1992.

[43] P. Vanoostende, P. Six, J. Vandewalle, and H.J. De Man,“Estimation of typical power of synchronous CMOS circuits using a hierarchy of simulators”, *IEEE Journal of Solid-State Circuits*, vol. 28, no. 1, pp. 26–39, 1993.

[44] C. Huang, B. Zhang, A. Deng, and B. Swirski, ”The design and implementation of PowerMill,” *International Symposium on Low Power Design, Dana Point, CA*, pp. 105–110, April 1995.

[45] C.-Y. Tsui, R. Marculescu, D. Marculescu, and M. Pedram,”Improving the efficiency of power simulators by input vector compaction”,*ACM/IEEE Design Automation Conference*, pp. 165–168, June 1996.

[46] R. Radjassamy, and J.D. Carothers, ”A Fractal Compaction Algorithm for Efficient Power Estimation”, *International Conference on Computer Design: VLSI in Computers and Processors*, pp. 542–547, 1998.

[47] C-Y. Hsu, W-Z. Shen,“Vector compaction for power estimation with grouping and consecutive sampling techniques”, *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 472–475, 2002.

[48] R. Tjarnstrom, “Power dissipation estimate by switch level simulation,” *IEEE International Symposium on Circuits and Systems*, pp. 881–884, May 1989.

[49] G. Y. Yacoub, and W. H. Ku, “An accurate simulation technique for shortcircuit power dissipation based on current component isolation,” *IEEE International Symposium on Circuits and Systems*, pp. 1157–1161, 1989.

[50] F. N. Najm, and M. G. Xakellis, ”Statistical estimation of the switching activity in VLSI circuits,” *VLSI Design*, vol. 7, no. 3, pp. 243-254, 1998

[51] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, “Testability Measures in Pseudorandom Testing”, *IEEE Transactions on CAD*, vol. 11, pp. 794–800, June 1992.

[52] B. Kapoor, “Improving the Accuracy of Circuit Activity Measurement”, *Proc. ACM/IEEE Design Automation Conference*, pp. 734–739, June 1994.

[53] C.-Y. Tsui, M. Pedram and A. M. Despain,“Efficient Estimation of Dynamic Power Dissipation with an Application”, *Proc. ACM/IEEE Design Automation Conference*, pp. 224–228, November 1993.

[54] P. Schneider, and U. Schlichtmann, “Decomposition of Boolean Functions for Low Power Based on a New Power Estimation Technique”, *Proc. 1994 Int’l Workshop on Low Power Design*, pp. 123–128, April 1994.

[55] P. H. Schneider, U. Schlichtmann, and B. Wurth, “Fast power estimation of large circuits“, *IEEE Design & Test of Computers*, vol. 13, no. 1, pp. 70–78, Spring 1996.

[56] R. Marculescu, D. Marculescu, and M. Pedram, ”Switching Activity Analysis Considering Spatiotemporal Correlations”, *Proc. 1994 Intl. Conference on Computer Aided Design*, pp. 294–299, November 1994.

[57] D. Sylvester, and H. Kaul, ”Power-driven challenges in nanometer design”, *IEEE Design & Test of Computers*, vol. 18, no. 6, pp. 12–21, 2001.

[58] R. Y. Rubinstein, ”Simulation and the Monte Carlo Method”, John Wiley & Sons, 1981.

[59] L. Wei, Z. Chen, M. Johnson, K. Roy, K and V. De, “Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits”, *Design Automation Conference*, pp. 489-494, 1998.

[60] F. Assaderaghi, D. Sinitsky, S.A. Parke, J. Bokor, P.K. Ko and C. Hu, ”Dynamic threshold-voltage MOSFET (DTMOS) for ultra-low voltage VLSI” , *IEEE Transactions on Electron Devices*, vol. 44(3), pp. 414-422, 1997.

[61] D. Nguyen, A. Davare, M. Orshansky, D. Chinnery, B. Thompson, and K. Keutzer, ”Minimization of Dynamic and Static Power Through Joint Assignment of Threshold Voltages and Sizing Optimization”, *International Symposium on Low Power Electronics and Design*, pp. 158-163, Aug. 2003.

[62] W. Hung, Y. Xie, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, Y. Tsai, ”Total Power Optimization through Simultaneously Multiple-VDD Multiple-VTH Assignment and Device Sizing with Stack Forcing”, *International Symposium on Low Power Electronics and Design*, 2004.

[63] A. Srivastava, D. Sylvester and D. Blaauw, “Power minimization using simultaneous gate sizing, dual-Vdd and dual-Vth assignment,”, *Design Automation Conference*, pp. 783-787, 2004.

[64] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda, and D. Blaauw, ”Duet: An Accurate Leakage Estimation and Optimization Tool for Dual Vt Circuits”, *IEEE Transactions on VLSI*, pp. 79-90, 2002.

[65] Q. Wang and S. Vrudhula, “Static power optimization of deep submicron CMOS circuit for dual Vt technology”, *International Conference on Computer-Aided Design*, pp. 490-496, 1998.

[66] M. Ketkar, and S. Sapatnekar, ”Standby Power Optimization Via Transistor Sizing and Dual Threshold Voltage Assignment”, *Proceedings of the International Conference on Computer Aided Design*, pp. 375-378, 2002.

[67] L. Wei, Z. Chen, M. Johnson, K. Roy, K and V. De, “Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits”, *Design Automation Conference*, pp. 489-494, 1998.

[68] F. Assaderaghi, D. Sinitsky, S.A. Parke, J. Bokor, P.K. Ko and C. Hu, ”Dynamic threshold-voltage MOSFET (DTMOS) for ultra-low voltage VLSI” , *IEEE Transactions on Electron Devices*, vol. 44(3), pp. 414-422, 1997.

[69] A. Abdollahi, F. Fallah, and M. Pedram, ”Runtime mechanisms for leakage current reduction in CMOS VLSI circuits”, *International Symposium on Low Power Electronics and Design*, pp. 419-424, Aug. 2002.

[70] D. Duarte, Y-F Tsai, N. Vijaykrishan, and M.J. Irwin, ”Evaluating run-time techniques for leakage power reduction”, *Proceedings of the ASP-DAC*, pp. 31-39, 2002.

[71] P. Pant, V. De and A. Chaterjee, ”Device-Circuit Optimization for Minimal Energy and Power Consumption in CMOS Random Logic Networks”, *Design Automation Conference*, Las Vegas, NV, June 1997.

[72] J. Kao, A. Chandrakasan, D. Antoniandis, ”Transistor Sizing Issues and Tool for Multi-Threshold CMOS Technology”, *Design Automation Conference*, Las Vegas, NV, June 1997.

[73] C. E. Shannon, “The Mathematical Theory of Communication”, *University of Illinois Press*, Urbana, IL, 1948.

[74] A. I. Khinchin, “Mathematical Foundations of Information Theory”, New York: Dover, 1957.

[75] J.E. Shore, “Relative entropy, probabilistic inference, and ai”, *Proceedings of the 2nd Annual Conference on UAI*, pp. 211-215, 1986.

[76] H. Chen, K. Roy and T-L. Chou, “Power sensitivity-a new method to estimate power dissipation considering uncertain specifications of primary inputs”, *International Conference on Computer Aided Design*, pp 40-44, 1997.

[77] M. Nemani and F. Najm, ”High-level power estimation and the area complexity of Boolean functions”, *International Symposium of Low Power Electronics and Design*, pp. 329-334, 1996.

[78] S. Bhardwaj, S. B. K. Vrudhula, D. Blaauw, ”τAU: Timing Analysis Under Uncertainty”, *Int’l Conference on Computer Aided Design*, pp. 615-620, 2003

[79] R.S. Ellis, “Entropy, Large Deviations, and Statistical Mechanics”, New York: Springer-Verlag, 1985.

[80] R. Burch, F. N. Najm, P. Yang, and D. Hocevar, ”Pattern Independent Current Estimation for Reliability Analysis of CMOS Circuits”, *Proceedings of the 25th Design Automation Conference*, pp. 294–299, June, 1988.

[81] Richard E. Neopolitan, “Learning Bayesian Networks”, Prentice Hall Series in Artificial Intelligence.

[82] Zhanping Chen, Mark Johnson, Liqiong Wei, and Kaushik Roy, ”Estimation of Standyby Leakage Power in CMOS Circuits Considering Accurate Modeling of Transistor Stacks”, *International Symposium on Low Power Electronics and Design*, pp. 239-244, August, 1998.

[83] J. P. Halter and F. Najm, ”A gate-level leakage power reduction methods for ultra-low-power CMOS circuits”, *IEEE Custom Integrated Circuits Conference*, pp. 475-478, 1997.

[84] S. Sirichotiyakul *et al.*, ”Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-Vt Circuits”, *Transactions on VLSI*, April, 2002.

[85] E. Acar *et al.*, ”Leakage and Leakage Sensitivity Computation for Combinational Circuits”, *International Symposium on Low Power Electronics and Design*, pp. 96-99, August, 2003.

[86] S. Bobba, I. N. Hajj, ”Maximum Leakage Power Estimation for CMOS Circuits”, *Proceedings of the IEEE Alessandro Volta Memorial Workshop on Low-Power Design*, 1999.