2004

Real delay graphical probabilistic switching model for VLSI circuits

Vivekanandan Srinivasan
University of South Florida

Real Delay Graphical Probabilistic Switching Model for VLSI Circuits

by

Vivekanandan Srinivasan

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:
November 1, 2004

Keywords: Bayesian Networks, Simulation, Inference, Sampling

DEDICATION

To My Parents.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my professor Dr. Sanjukta Bhanja for guiding me in this thesis. Dr. Bhanja's vision and guidance, has immensely helped me to stay the course and complete this work , and has also exposed me to new vistas in the research area. I would also like to express my gratitude to Dr. Wilfrido Moreno and Dr. Yun-Leei Chiou for agreeing to serve in my committee. I wish to thank my parents for the hardships they have endured to provide me with this education. I wish to thank my grandparents for all the support they have offered me, especially my grandmother, an intelligent and compassionate woman, she has been supportive of all my ventures. I would also like to thank my friends, in the VLSI Design Automation and Test Lab, Bheem, Nirmal,Saket, Sathish, Shiva and Tara, for crucial help and suggestions throughout the course of my work. I would also like to thank Ponraj and Bodka for suggestions in Coding area.

TABLE OF CONTENTS

LIST OF TABLES iii

LIST OF FIGURES iv

ABSTRACT v

CHAPTER 1 INTRODUCTION 1
- 1.1 Low Power VLSI Design 1
- 1.2 Static Power Dissipation 2
- 1.3 Dynamic Power Dissipation 2
- 1.4 Contributions of this Thesis 4
  - 1.4.1 Graphical Representation 4
  - 1.4.2 Bayesian Networks 4
  - 1.4.3 Gate Delay Modeling 5
- 1.5 Flow of this Thesis 6

CHAPTER 2 PRIOR WORK 8

CHAPTER 3 BAYESIAN NETWORKS 12
- 3.1 Bayesian Networks 12
- 3.2 Functionality of LIDAG BN 19

CHAPTER 4 REAL DELAY PROBABILISTIC MODEL 21
- 4.1 Types of Delay Model 22
  - 4.1.1 Zero Delay Model 22
  - 4.1.2 Unit Delay Model 22
  - 4.1.3 Variable Delay Model 22
- 4.2 Gate Delay Model 23
- 4.3 Expanded Circuit Representation 24
- 4.4 Real Delay Probabilistical Model 26
  - 4.4.1 Dependencies in Primary Inputs 26
  - 4.4.2 Higher Order Spatio-Temporal Dependencies 28
  - 4.4.3 Higher Order Temporal Dependencies 29

CHAPTER 5 INFERENCE TECHNIQUES 31
- 5.1 Exact Inference 32
  - 5.1.1 Polytree Algorithm 32
  - 5.1.2 Clustering 32

5.2 Approximate Sampling Algorithms 32
5.2.1 Stochastic Sampling Algorithms 32
5.2.2 Probabilistic Logic Sampling 33
5.2.3 Importance Sampling 34
5.2.4 Adaptive Importance Sampling 35
5.2.5 Evidence Pre-propagation Importance Sampling Algorithm 35

CHAPTER 6 RESULTS 37

CHAPTER 7 CONCLUSION 40
7.1 Future Work 41

REFERENCES 42

LIST OF TABLES

Table 3.1.	Conditional Probability Specifications for the Output and the Input Line Transitions for Two Input NAND Gate.	20
Table 4.1.	Conditional Probability Specifications for Primary Input Dependency.	28
Table 4.2.	Conditional Probability Specifications for Output Nodes.	30
Table 6.1.	Maximum Time Instants for Some Combinational Benchmark Circuits.	37
Table 6.2.	Switching Activity Estimation Error Statistics Based on Delay DAG Modeling, Using PLS Inference Scheme, Using 1000 Samples, for ISCAS'85 Benchmark Combinational Circuits.	38

LIST OF FIGURES

Figure 1.1.	Moore’s Law.	2
Figure 3.1.	2-Input OR Gate.	13
Figure 3.2.	C17 Benchmark Circuit.	14
Figure 3.3.	Bayesian Network of C17 Benchmark Circuit.	15
Figure 4.1.	Glitch Generation.	21
Figure 4.2.	Variable Delay.	24
Figure 4.3.	Expanded Circuit.	25
Figure 4.4.	DAG of A Real-Delay Model.	27
Figure 5.1.	Flowchart.	33

REAL DELAY GRAPHICAL PROBABILISTIC SWITCHING MODEL FOR VLSI CIRCUITS

Vivekanandan Srinivasan

ABSTRACT

Power optimization is a crucial issue at all levels of abstractions in VLSI Design. Power estimation has to be performed repeatedly to explore the design space throughout the design process at all levels. Dynamic Power Dissipation due to Switching Activity has been one of the major concerns in Power Estimation. While many Simulation and Statistical Simulation based methods exist to estimate Switching Activity, these methods are input pattern sensitive, hence would require a large input vector set to accurately estimate Power. Probabilistic estimation of switching activity under Zero-Delay conditions, seriously undermines the accuracy of the estimation process, since it fails to account for the spurious transitions due to difference in input signal arrival times. In this work, we propose a comprehensive probabilistic switching model that characterizes the circuit's underlying switching profile, an essential component for estimating data-dependent dynamic and static power. Probabilistic estimation of Switching under Real Delay conditions has been a traditionally difficult problem, since it involves modeling the higher order temporal, spatio-temporal and spatial dependencies in the circuit. In this work we have proposed a switching model under Real Delay conditions, using Bayesian Networks. This model accurately captures the spurious transitions, due to different signal input arrival times, by explicitly modeling the higher order temporal, spatio-temporal and spatial dependencies. The proposed model, using Bayesian Networks, also serves as a knowledge base, from which information such as cross-talk noise due to simulataneous switching at input nodes can be inferred.

CHAPTER 1

INTRODUCTION

1.1 Low Power VLSI Design

In 1963, CMOS circuits were invented as a low power alternative to TTL, today, CMOS is the predominant technology in digital integrated circuits albeit increased functionality, power is still one of the major concerns of the VLSI design community. The quest for smaller devices with increased functionality has driven the CMOS technology into the deep submicron region. The shrinking gate length has lead to an exponential increase in the number of transistors per unit area. While this increase in device density has created smaller and portable electronic devices, which are easier to market, it has also lead to an exponential increase in the power consumption and power dissipated per unit area. The increase in power dissipation requires higher cooling costs and a higher battery weight, which adversely affect the marketability of an electronic device. This explains the emergence of low power design as one of the key research areas in VLSI design.

Intel founder Dr. Gordon Moore, states that semiconductor denstiy, and hence performance, doubles roughly every 18 months. Gordon Moore's insightful observation on device advancement has had a direct correlation with power dissipation. [Figure 1.1.] shows the power dissipation of some representative Intel microprocessors as a function of time. The scatter in data clearly suggests technology advancements to reduce power. But the overall trend is the exponential increase in power dissipation with time.

Power dissipation is factored into the following components:

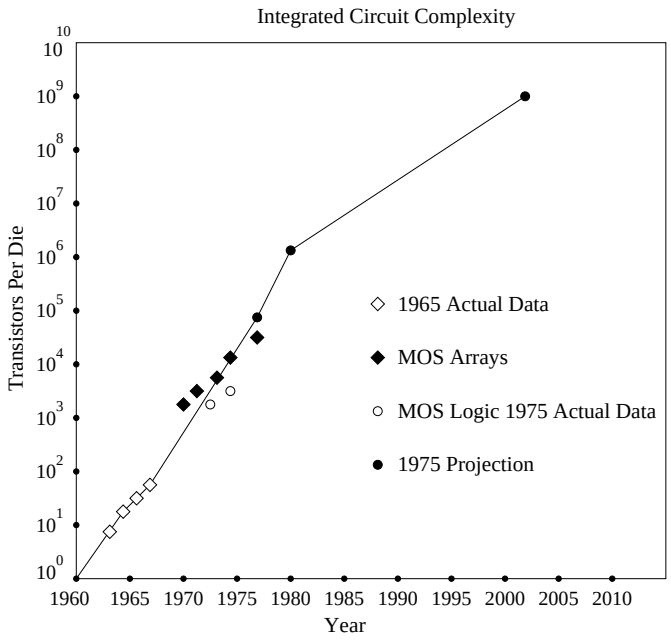


Figure 1.1. Moore's Law.

1.2 Static Power Dissipation

Static Dissipation occurs due to reverse bias leakage between diffusion regions and the substrate. Subthreshold conduction also contributes to static dissipation. Leakage current is gaining significance as we are speeding into the nano region.

1.3 Dynamic Power Dissipation

Short-circuit power Dissipation occurs when there is a short current pulse from Supply to Ground. The direct path from Supply to Ground is due to the p-type and n-type transistors being 'on' for a brief period of time during transition from '1' to '0' or from '0' to '1'.

Dynamic Dissipation also occurs due to the charging and discharging of parasitic capacitance during switching. Our quest to produce faster devices, thus increasing the clock frequency, has bestowed even greater importance in this component of power dissipation.

Average Dynamic power dissipation at a gate is given by,

$$P_{dynamic} = \sum_i C_i V_i^2 f_i \tag{1.1}$$

Where Vdd is the supply voltage , $fclk$ is the clock frequency, Cload is load capacitance and $sw(x)$ is the average switching activity of the output node x.

The factors affecting power dissipation being supply voltage, load capacitance and switching activity. Supply voltage and clock frequency are known to the designers. Supply voltage, by virtue of it's quadratic dependence on power dissipation, has been an effective tool in minimizing power dissipation.

IC power management requires the inclusion of power along with timing and area, as parameters managed throughout the design process.Pre-emption has been the only option for low power designers. Estimation of power has been performed at various levels of abstraction, namely behavioral level, RT level, gate level, circuit level etc. Optimization is performed at each level before synthesizing to lower levels. The higher the level of abstraction, faster is the speed of estimation but the accuracy is lower. This work is focused at the gate level of power estimation. At this level, gate capacitances are estimated from the knowledge of the logic structure, the challenge is to estimate switching activity .

Estimation of switching activity is a challenging process, input statistics, correlation between nodes, gate type and gate delays are some of the factors that affect switching activity.

Switching activity at a circuit node, can be defined as the average number of transitions at that node per unit time. To calculate switching activity the knowledge of the previous and present state of a node is required, this helps in modeling for temporal correlation. Reconvergent fanouts and any dependencies between the primary inputs in the circuit gives rise to spatial correlation . For better accuracy the spatial and temporal correlations should be taken into account in the model.

First order Temporal and higher order Spatial correlations are accurately captured in the work of bhanja et al. but their work fails to account for gate delay. For accurate estimation of power, gate delay has to be taken into account, signals arriving at the internal nodes of gate, might arrive at different time instances, causing spurious transitions to take place. Spurious transitions, also called glitches, dissipate power, this has to be taken into account inorder to accurately estimate power. Higher order temporal correlations, along with spatiotemporal correlations, has to be taken into account for a model capturing

spurious transitions along with the desired switching activity. The main focus of this work is to design a model that would take into consideration these higher order temporal correlations, and thereby account for gate delay to accurately estimate Power dissipation.

1.4 Contributions of this Thesis

Probabilistic modeling of Gate delay has been traditionally a difficult problem. This work suggests a novel switching model that would capture all the spurious transitions arising due to gate delay, by explicitly modeling for higher order temporal correlations. The switching model is novel in that, it could also be used to estimate the probability of cross-talk noise between any two nodes, and as observed in the work of Ramani *et al* it could also be used to estimate the nodes with high probability of leakage.

It has been shown that Bayesian networks elegantly capture the first order temporal and spatial correlations with reduced computation time. This work models the temporal correlation by explicitly modelling the dependencies between different time instances of a node and thereby increasing the accuracy in the estimation process.

1.4.1 Graphical Representation

Inferencing judgement from Probabilistic models involves defining a joint distribution function on all propositions and their combinations. Constructing a joint distribution function for n variables, would require a table with 2^n entries. This leads to an exponential growth in the complexity of calculation along with an increase in the size of the circuit, also the human mind does not reason with such heavy computations. Hence, Graphical Representation is used, which apart from making inference of desired results easier, also facilitates the verification of existing dependencies and investigation of new dependencies.

1.4.2 Bayesian Networks

Bayesian Network has been used in this work to probabilistically model the switching activity. Bayesian Networks are Directed Acyclic Graphs(DAGs), in which the variables are represented as nodes, and the causal influence between the variables are represented by directed arcs, the effect a parent node

has on its children is represented by a conditional probability table. By introducing directionality in the graphs, Bayesian Networks are capable of displaying induced and non-transitive dependencies.

Logic Induced Directed Acyclic Graph(LIDAG) is constructed based on the logical structure of the circuit. Conditional probability tables, which indicates the strength by which parent nodes affect a child node, represent the switching activity and are mapped one-to-one, onto a Bayesian Network thus preserving the dependency of the probability function. Each node in the LIDAG, represents a signal which can have four possible states, $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$, $1 \rightarrow 1$. Directed edges are drawn from nodes representing the signals to their respective fanouts, the conditional probability table reflects the attribute of the node and determines how the switching input from the parent nodes affect the node. The work of Bhanja et al proves that a LIDAG, thus obtained, is a Bayesian Network which is a minimal representation that captures all the independency relationships in the circuit.

1.4.3 Gate Delay Modeling

There are various types of gate delay models, zero delay, unit delay and variable delay are a few among them. In this work we construct a variable delay model, assuming delays of the gate to be proportional to their fanout.

Inputs arriving at the gate at different time instants would cause undesired or spurious transitions at the output. These spurious switchings are accounted for, by explicitly modelling for higher order temporal and spatiotemporal correlations. We instantiate a gate for each time instant a possible spurious transition can occur. A logic gate at a particular time instant could be affected only by the instances of inputs with time instants immediately preceding the time of the logic gate.

In the zero gate delay of Bhanja et al, by decoupling delay from the model, they assume instantaneous transmission of signals, which is not the real case, there is always some delay in the arrival of signals, thus hampering the accuracy of power estimation process. In the work of Manich *et al* [9], they use this gate delay model to find the couple of vectors that would maximize the weighted switching activity, their work uses a simulation model. A large number of vectors should be used for gaining accuracy, thus making the process a time complex one . Our work uses a probabilistic model, that would give higher accuracy in a smaller run time. As said earlier, the model uses higher order to account for glitches we

would need to have anterior transition and present transition as opposed to anterior and present states, Hence the model would have the knowledge of the gate states at time instants, $t2, t1, t,$, where t represents the present state, and all other time instants till it reaches a stable desired value.

The probabilistic model explicitly captures the temporal correlation between all the different time instants, this is done by drawing edges, from the appropriate time instances of parent nodes, to the time instance of child node, it would affect. The switching probabilities of a node would denote the probability of switching, from it's state corresponding to previous input transition probability to present input transition probability, this switching probability of the child node is of no consequence in this model, we would have to compare the switching between successive time instances of the child node to get the switching probability between these time instances. Let's say a node A at time instant 2 switches from $(0 \rightarrow 0)$ $(t-1 \rightarrow t)$ and the same node A at time instant 3 switches from $(0 \rightarrow 1)$ $(t-1 \rightarrow t)$, we should compare the states at time instant t between the instances 2 and 3 of node A, in this case the node has switched from $(0 \rightarrow 1)$, thus we would have to calculate the sum of switching probability between all the successive time instances of node A to obtain the final switching probability of node A. Switching probability thus calculated, captures the effect, the previous input leaves behind in the form of spurious transitions.

The switching model depicted in this work is novel in that, it accounts for gate delay. The switching model apart from accounting for gate delay, can be used to identify the nodes with high probability of leakage, by using the method observed in the work of Ramani [1]. Some of the earlier works in this related field, have failed to account for dependencies in primary input, this could be taken care of in this model by incorporating the vectorless approach, suggested by Ramalingam [2]. Hence by incorporating the above features, this model apart from being accurate, would be comprehensive, as it could identify nodes with high probability of leakage and cross-talk noise along with the switching activity.

1.5 Flow of this Thesis

The literature review is presented in Chapter 2. Probabilistic Model is discussed in Chapter 3. Probabilistic model under Gate Delay conditions is discussed in Chapter 4. Bayesian Inference Techniques is

presented in Chapter 5. The Results and future work that could be carried on with this model is discussed in Chapter 6.

CHAPTER 2

PRIOR WORK

Shrinking Device structures have continued to cause an increase in the device density and hence an increased power density. Power estimation has been performed at every stage in design process, to aid designers in optimizing the design to reduce Power consumption. Despite shrinking gate lengths and various changes in technology, Dynamic power dissipation continues to be one of the major issues in Power estimation. Switching activity is an important parameter in the calculation of Dynamic power estimation. A novel term 'transition density' was used by Najm in his work [18], to estimate average switching rate. *Definition:* [18] The transition density of a logic signal $x(t)$,$t \in (-\infty, +\infty)$, is defined as

$$D(x) = lim_{T\to\infty}\frac{n_x(T)}{T} \tag{2.1}$$

where $n_x(T)$ represents the number of transitions in $x(t)$ in the time interval $(-T/2, +T/2)$.

The expression for calculating average power would be ,

$$P_{av} = \frac{1}{2}CV_{dd}^2 lim_{T\to\infty}\frac{n_x(T)}{T} \tag{2.2}$$

Transition densities are given at the primary inputs and are propagaged into the circuit to get the transition densities at the output and internal nodes. This method assumes the inputs are independent and hence fails to account for the correlations in the input nodes. Bhanja *et al* [4] introduced Bayesian Networks for switching estimation in combinational circuits. In their work switching activity was represented as

$$Sw(X) = P(X_{0\to1}) + P(X_{1\to0)} \tag{2.3}$$

where $X_{0\to1}$ and $X_{1\to0}$ denotes a signal transition from 0 to 1 and 1 to 0 at node X respectively. 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 is affected by the following types of correlations,

1. Temporal Correlation represents the correlation between the previous value of a signal and present value of a signal 2. Spatial Correlation represents the correlation caused by re-convergent fanouts and dependencies in input. 3. Spatio-Temporal Correlation, a combination of spatial and temporal correlation, represents the dependence of a signal to the previous value of a spatially connected signal. 4. Sequential represents the spatial correlation due to feedback state lines.

Power estimation is performed at various levels of design process. Behavioural level power estimation [76, 77, 78, 80, 81]is a high level of power design, this is the earliest stage in design process, design changes at this stage are flexible and saves design time. Register Transfer Level(RTL) [82, 83, 84, 85, 86] power estimation is done when the circuit is still expressed as blocks, with each block having it's own combinational circuit. High level power estimation though flexible and saves design time, is inaccurate. Gate level Power estimation though costly in terms of design time provides greater accuracy. In this work we estimate power at Gate level.

Simulation [45, 71, 53, 60] methods are the simplest method to estimate power, these methods while being accurate, consume a lot of time. The simulation methods are pattern dependent. In the design process, the designers do not have knowledge of the nature of inputs, this hampers accuracy.

Statistical simulation methods are similar to simulation models, they differ in that, the input vectors are selected based on user defined probabilities, also the accuracy and confidence can be specified by the user. [25, 70, 47, 72, 73, 74, 75] have suggested some methods using Statistical simulation. While these methods provide an improvement in time over Simulation methods, their accuracy depends on their knowledge of inputs.

Probabilistic simulation involve the construction of special circuit models based on the probabilistic method used. The computation time is fast and tractable. The probabilistic methods provide a compact way of representing the logic signals, these probability values are propagated through the model, based on the influence of these values on the gates, power is calculated. Due to the probabilistic nature of the

inputs, changes in input pattern can be easily modelled. The dependencies in the circuit has to be taken care of in probabilistic propagation to get an accurate model.

Parker and McCluskey [21] presented one of the earliest methods that used signal probability for probabilistic propagation. The estimate obtained was grossly inaccurate as the temporal and spatial correlation were neglected.

Binary Decision Diagrams(BDD) for signal proability was first proposed by Chakravarti *et al* [16] , In this method, the signal probability at the output is calculated by builiding OBDD corresponding to the function of the node in terms of circuit inputs , and then performing a postorder traversal of the OBDD using equation

$$P(Y) = P(x)P(f_x) + P(\overline{x})P(f_{\overline{x}}) \tag{2.4}$$

Ercolani *et al* [19] describe a procedure for propagating signal probabilities from circuit input towards output using only pairwise correlation between circuit lines. This method ignores higher order correlations.

The above methods ignored to take into account the gate delay, ignoring the power dissipation due to hazards and glitches.

Ghosh *et al* [31] estimate average power dissipated in combinational and sequential circuits, using a general delay formula. They use symbolic simulation to produce a set of boolean function that represent the condition for switching at different time points for each gate. From, the input switching rate, probability of each gate switching at any time point is calculated. The sum of switching activity in the entire circuit over all the time points for all the gates corresponding to a clock cycle is calculated. The major disadvantage of this method is that for medium to large circuits the symbolic formulae become too large to build.

Burch *et al* [25] introduced the concept of probability waveform. The waveform consists of transition edges or events over time from the inital steady state time,0^- to final steady state time, ∞, where each event is annotated with an occurence probability. The probability waveform represents all possible logical waveforms of that node. Given these waveforms, switching activity of x, which includes hazards

and glitches is calculated,

$$E_x(sw) = \sum_{i \in eventlist(x)} (p(x^t_{0\to 1}) + p(x^t_{1\to 0}). \tag{2.5}$$

Najm *et al* [24]propagates transition waveforms from circuit inputs throughout the circuit and estimates total power consumption. This method does not take into spatial correlations due to reconvergence.

Tsui *et al* [23] proposed a tagged probabilistic simulation approach, in this method logic waveforms at a node are broken into four groups, each group being characterized by it's steady state values. Each group is then combined into a probability waveform with appropriate steady-state tag. Given the tagged probability waveforms at the input of node n, it is possible to compute tagged probability waveforms at the output. The correlation between probability waveforms at inputs is approximated by correlation between the steady state values of these lines, which is calculated by describing the node function in terms of some set of intermediate variables in the circuit. This method does not take into account the slew in the waveforms, the higher order spatial correlations are also not modelled in this method.

Najm *et al* [17] introduced the concept of transition density, D, the transition densities are propagated throughout the circuit. Transition density of each node is calculated as follows:

$$D(y) = \sum_{i=1}^{n} P(\frac{\partial y}{\partial x_i} D(x_i)) \tag{2.6}$$

The disadvantage of this method is that it assumes independence in the input function.

The work of Bhanja et al [4] while capturing higher order spatial and first order temporal correlation gives an accurate switching estimation for zero-delay models, but fails to account for delay of logic gates, thus failing to account for accurate switching estimation of the combinational circuits.

In this work we propose a Probabilistic method using Bayesian Networks, to accurately model any order of Temporal and Spatio-Temporal correlation, along with higher order Spatial correlation, in the process we infuse *dynamic* tendencies in the Bayesian Network. This enables the model to capture all desired and undesired transitions that a Gate ouptut expereiences. Thus, enabling an accurate Probabilistic switching estimation model under Real Delay conditions.

CHAPTER 3

BAYESIAN NETWORKS

Graphical structures have been used in this work to estimate the switching probabilities of VLSI circuits. Graphical structures facilitates an intuitive understanding of the dependencies among the nodes in a model, as against the traditional method in which we would have to verify if $P(x,y) = P(x).P(y)$, to ascertain the dependency between the variables x and y.

The essence of Graphical representations, is to represent probabilistic inference in the context of human reasoning and to avoid complex numerical calculations. Graphical representations, preserve and highlight, the dependencies in the circuit impervious to changes in the numerical input. In our case, we represent combinational circuits as graphs, it is comparitively an easier job, since we have a sense of a nodes direct and indirect neighbours. In this work we have represented the circuits as Bayesian Networks and we would discuss in detail about these networks in this chapter. This chapter follows the flow of PEARL's [40], with probabilistic models representing combinational circuits, to lucidly explain the nature of Bayesian Networks and to prove that our model works.

3.1 Bayesian Networks

Bayesian networks are Directed Acyclic Graph(DAG) structures i.e., the nodes are connected by directed arcs and there is no closed connection while traversing the directed arrows (acyclic). The importance of Directed networks as opposed to undirected networks can be explained by a 2-input OR gate. Let A and B be the inputs at this *OR* gate and let C be the output. The undirected graph and the directed graph is shown in 3.1.. From the undirected graph we infer that A and B are independent given C. But in reality if the value of C is known, say $'1'$, learning the value of A say $'0'$, would affect the value of B. If we add a link between A and B, it would indicate that A and B are no longer independent of each other. The Bayesian Network of the logic gate, denotes that A and B are independent if the value of C is not

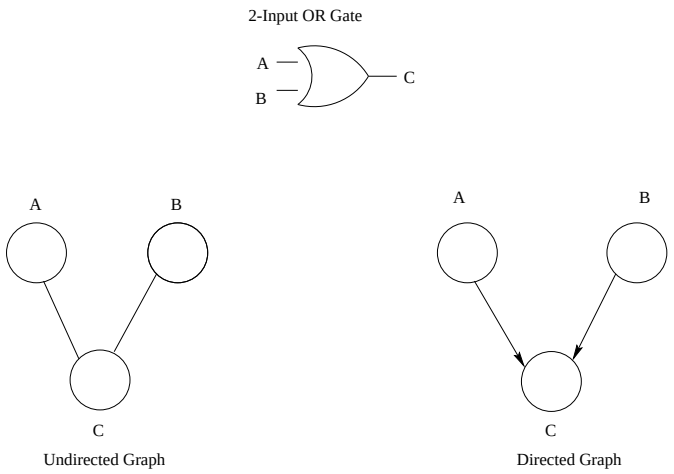


Figure 3.1. 2-Input OR Gate.

known. Once C is instantiated, A and B are no longer independent. This induced dependency is captured, in the Bayesian Network shown in 3.1., by virtue of directed connectivity, called *dseparation.* Directed graph representations make it easy to quantify the links with local, conceptually meaningful parameters that turn the network as a whole into globally consistent knowledge base.

It would be only appropriate to stress, the importance of independencies, since the whole network has been weaved around this concept. A node would act only based on the beliefs of it's neighbours, and would discount the effect of all other nodes it is independent from. This is the core concept that makes the network appealing. Hence, care has to be taken to capture all the independencies in the network. Henceforth, in this chapter, we would prove that our model satisfies all the theorems and definitions quantifying a Bayesian network.

Markov networks, due to lack of directionality states that, if the removal of some subset Z of nodes from the networks render nodes X and Y disconnected, then X and Y are independent. This was explained above with the 2-input OR gate example.The following definition from [40], gives the criterion for seperation in a DAG.

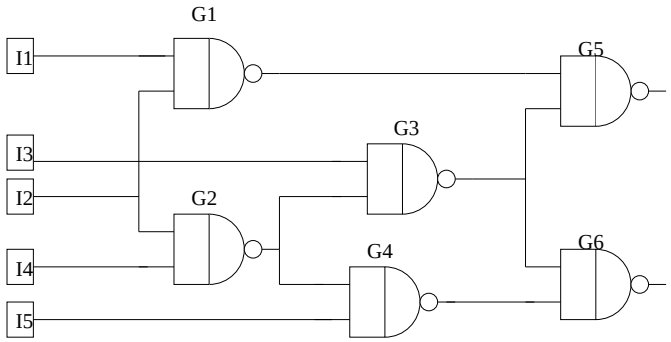


Figure 3.2. C17 Benchmark Circuit.

DEFINITION:1 If X, Y, and Z are three disjoint subsets of nodes in a DAG D, then Z is said to *d-separate* X from Y, denoted $< X|Z|Y >_d$, if along every path between a node in X and a node in Y there is a node w satisfying one of the following two conditions: 1. w has converging arrows and none of w or its descendants are in Z. If, on the contrary, w or its descendents is in Z, the path is said to be activated by Z. 2. w does not have converging arrows and w is in Z. In this case, the path is said to be blocked by Z.

The combinational benchmark circuit, $C17$ is given in, 3.2..

The bayesian network of a benchmark circuit, $C17$, 3.3. has been used here to explain this seperation criterion.

In the bayesian model of combinational circuit, $C17$ represented above, nodes represent the logic gates and the primary inputs represent the primary inputs of the circuit, the dependency between the nodes are represented by directed arrows.

Consider nodes, $G2, G3$ and $G6$. $G3$ lies in the path of $G2$ and $G6$, and $G3$ does not have any converging arrows, therefore $G3$ according to the above definition blocks the path from $G2$ to $G6$, and therefore d-separates $G2$ and $G6$. This type of separation is similar to the cutset separation in undirected graphs.

Consider the nodes $G1, G5$ and $G3$. These nodes if denoted by undirected graphs, would lead us to believe that, $G1$ and $G3$ are independent given $G5$, but this is an incorrect assumption, in reality, $G5$ denotes the output of a nand gate, given the value of $G5$, knowing the state of $G1$ would affect the state of $G3$. As in the above bayesian model, when directionality and a clear d-separtion definition is taken

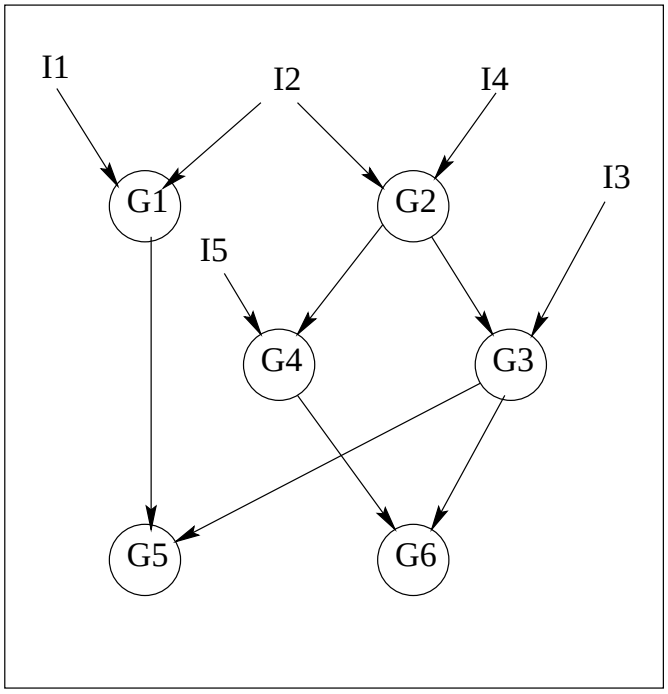


Figure 3.3. Bayesian Network of C17 Benchmark Circuit.

into consideration, we understand the path from $G1$ to $G3$ is activated by $G5$, and that learning the state of $G5$, makes $G1$ and $G3$ dependent on each other.

The mathematical equivalent of independency criterion is given by the following definition.

DEFINITION:2 Let $U = A, B, \ldots$ be a finite set of variables with discrete values. Let $P(.)$ be a joint probability function over the variables in U, and let X, Y and Z stand for any three subsets of variables in U. X and Y are said to be conditionally independent given Z if

$$P(X|Y,Z) = P(X|Z) whenever P(Y,Z) > 0 \tag{3.1}$$

and this independency is denoted by $I(X,Z,Y)$ meaning, Given the state of Z, X and Y are rendered independent.

The independency of the node $G6$ from $G2$ given $G3$, in the above figure, could be verified by the equation,

$$P(G6|G2,G3) = P(G6|G3) whenever P(G2,G3) > 0 \tag{3.2}$$

The independency of nodes is denoted by $I(G2, G3, G6)$.

The Numerical representation of probabilities may not shed light on induced dependencies, Axiomatization of probabilities allows us to intuitively conjure new theorems, that would better help us to model dependencies. In a Probabilistic model, an independence relationship $I(X,Z,Y)$ must satisfy the following axioms according to PEARL [40],

SYMMETRY

$$I(X,Z,Y) \Longleftrightarrow I(Y,Z,X) \tag{3.3}$$

The symmetry axiom states that for any given state of Z, if learning the state of X tells us nothing new about Y, then learning the state of Y would tell us nothing new about X

DECOMPOSITION

$$I(X,Z,YUW) => I(X,Z,Y)\&I(X,Z,W) \tag{3.4}$$

Decomposition axiom states that given the state of Z, if learning the state of both, Y and W tells us nothing new about X, then learning Y and W separately would also tell us nothing new about X

WEAK UNION

$$I(X,Z,YUW) => I(X,ZUW,Y) \tag{3.5}$$

Weak union axiom states that learning irrelevant information W cannot help irrelevant information Y become relevant to X.

CONTRACTION

$$I(X,Z,Y)\&I(X,ZUY,W) => I(X,Z,YUW) \tag{3.6}$$

Contraction axiom states that if we judge W irrelevant to X after learning some irrelevant information Y, then W must have been irrelevant before we learned Y.

A Dependence model, M of a domain should capture all the conditional independencies, $I(X,Z,Y)$ amongst the variables in that domain.

Definition:3 An undirected graph G is a dependency map or$D-map$ of M if there is a one-to-one correspondence between the elements of U and the nodes V of G, such that for all disjoint subsets X,Y,Z

of elements we have

$$I(X,Z,Y)_m => < X|Z|Y >_g \tag{3.7}$$

Similarly, G is an independency map $or I-map$ of M if

$$I(X,Z,Y)_m <= < X|Z|Y >_g \tag{3.8}$$

G is said to be a perfect map of M if it is both a $D-mapandanI-map$.

DEFINITION:4 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

$$< X|Z|Y >=> I(X,Z,Y)_m. \tag{3.9}$$

A DAG is a *minimal I-map* of M if none of its arrows can be deleted without destroying its I-mapness.

The figure represents a minimal I-map, since it represents all the conditional independencies and destroying any edge would destroy its I-mapness.

DEFINITION:5 Given a probability distribution P on a set of variables U, a DAG D = U,E is called a Bayesian network of P if D is a minimal I-map of P.

Since our figure is a minimal I-map it is indeed a Bayesian Network.

DEFINITION:6 A Markov blanket BL1(A) of an element A *belongsto* U is any subset S of elements for which

$$I(A,S,U-S-A) and (A \notin S). \tag{3.10}$$

A set is called a Markov boundary of A, denoted B1(A), if it is a minimal Markov blanket of A, i.e., none of its proper subsets satisfy the above equation.

THEOREM:1 Every element $AbelongstoU$ in a dependency model satisfying symmetry, decomposition, intersection, and weak union has a unique Markov boundary B1(A). Moreover, B1(A) coincides with the set of vertices Bg0(A) adjacent to A in the minimal I-map G0.

DEFINITION:7 Let M be a dependency model defined on a set $U = X_1, X_2, \ldots, X_N$ of elements, and let d be an ordering $(X_1, X_2, \ldots, X_i, \ldots)$ of the elements of U. The boundary strata of M relative to d is an ordered set of subsets of U, $(B_1, B_2, \ldots, B_i, \ldots)$, such that each Bi is a Markov boundary of Xi with respect to the set $U_i = X_1, X_2, \ldots, X_{i-1}$, i.e., B_i is a minimal set satisfying $B_i \subset U_i$ and $I(X_i, B_i, U_i - B_i)$. The DAG created by designating each B_i as parents of vertex X_i is called a boundary DAG of M relative to d.

The boundary DAG for the figure[3.3.], representing a C17 combinational circuit[3.2.] is done such that a nodes markov boundaries (in our case a node's parents) are created before the creation of node, this is understood intuitively, as it reflects the structure of the combinational circuit. The boundary strata, B_M of 3.3. is given as

$$B_M = \{\phi, \phi, \phi, \phi, \phi, I1, I2, I2, I4, I5, X2, X2, I3, X3, X4, X1, X3\} \tag{3.11}$$

THEOREM:2 [Verma 1986]: LetM be any semi-graphoid. If D is a boundary DAG of M relative to any ordering d, then D is a minimal I-map of M.

The definitions and theorems given above define the structure of Bayesian Network. It should be stated here that, the above Definitions, Axioms and Theorems , religiously follow PEARL, and comments have been added to explain them in our context.

Bayesian Networks were introduced for switching activity estimation in combinational circuits in the work of Bhanja et al. They proposed a LIDAG, Logic Induced Directed Acyclic Graph, to reflect the combinational circuits as Zero-delay model. Definition and Theorem from Bhanja et al. would be used to prove that the 3.3. is a LIDAG.

The nodes $\{G1, G2, \ldots, G6\}$ in the 3.3. represent gate outputs, and the nodes $\{I1, I2, \ldots, I5\}$ represent the primary inputs of the gate. The output of a gate depends on the inputs to the gate, which may be primary inputs or outputs from other gates, and the nature of it's conditional probability table. The switching of the input nodes and the gate output nodes are the random variables of interest,$\{I1, I2, \ldots, I5, G1, \ldots, G6\}$. Each node takes four possible states $\{x_{00}, x_{01}, x_{10}, x_{11}\}$ representing the probability of the switching transitions: $(0 \to 0)$,$(0 \to 1)$,$(1 \to 0)$ and $(1 \to 1)$.

DEFINITION:8 In a LIDAG the nodes represent the switching at gate output lines, the arcs represent the input lines on which the node depends, the arcs might originate from primary input nodes or other nodes representing gate output line. DAG shown in 3.3. corresponds to the C17 benchmark circuit.

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

Proof: From the definition of LIDAG above, it is understood that in the LIDAG structure the parents of each node are its Markov boundary elements, The LIDAG is a boundary DAG of M, which could be verified by EQUATION, hence from THEOREM 9 it follows that D is a minimal I-map of M.

3.2 Functionality of LIDAG BN

Let us try to construct a LIDAG BN for the benchmark C17 circuit , Each node is assigned four possible states,$(x_{00}, x_{01}, x_{10}, x_{11})$. $x_{a,b}$, denotes the probability of transition from previous value a at time, $t-1$ to present value b at time, t.$a, b \in \{0,1\}$ .By defining the states as switching probabilities, we have knowledge of the node's previous and present value, thus capturing first order temporal dependencies. The higher order spatial correlations are captured by the arcs between the dependent nodes and the nature of dependency is quantified by the conditional probability table. The DAG nature of the model helps us in intuitive identification of conditional independencies. This, facilitates the computation of the joint probability function, which can be expressed as a product of conditional probabilities.

$$P(x_1, \ldots, x_N) = (\prod_v P(x_v | x_{parent(v)})) \tag{3.12}$$

The joint probability distribution function for the LIDAG represented in figure[3.3.] is expressed by the following factored form.

$$\begin{aligned} P(x_1, \ldots, x_6) = {} & P(x_6|x_4, x_3) P(x_5|x_1, x_3) P(x_4|I5, x_2) P(x_3|x_2, I3) P(x_2|I2, I4) \\ & P(x_1|I1, I2) P(I1) P(I2) P(I3) P(I4) P(I5) \end{aligned} \tag{3.13}$$

Table 3.1. Conditional Probability Specifications for the Output and the Input Line Transitions for Two Input NAND Gate.

Two Input NAND gate					
$P(X_{output}\|X_{input1}, X_{input2})$ 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}

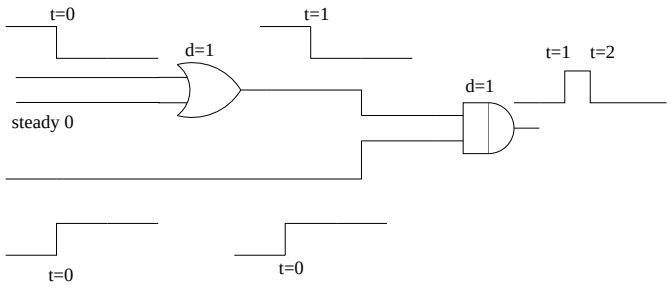
The conditional probability table reflects the nature of the gate. A complete conditional probability table for $P(x_6|x_4, x_3)$ is given by 4^3 entries, since each variable has 4 states.Conditional propability tables of a Nand is listed in Table[3.1.].

In this chapter we have discussed the nature and formation of Bayesian Networks in the next chapter we would discuss the construction of Real delay graphical probabilistic model.

CHAPTER 4

REAL DELAY PROBABILISTIC MODEL

Power Dissipation has been a major concern for VLSI designer for over a decade. Dynamic power dissipation has been and continues to be one of the major sources of power dissipation. The charging and discharging of load capacitances, due to switching, is one of the main factors of Dynamic power dissipation. The difference in arrival times of signals at a gate input, leads to spurious transitions, also called as glitches. Glitch generation shown in [Figure 4.1.] is due to signals travelling through gates with different delays, before arriving at the inputs. These spurious transitions also play a major role in power dissipation. Power estimation using Zero-delay models do not take these spurious transitions into considerations, thus grossly underestimating the power dissipation for circuits with unbalanced path lengths.



SPURIOUS TRANSITION AT A NODE [K. ROY]

Figure 4.1. Glitch Generation.

4.1 Types of Delay Model

4.1.1 Zero Delay Model

Zero Delay Model assumes that the gates have no delay, hence an instantaneous transmission of signals through gates is assumed. This assumption neglects the switching activity due to spurious transitions, hence it is inaccurate. This type of model can be represented probabilistically by taking into account first order temporal and spatio-temporal dependencies.

4.1.2 Unit Delay Model

Unit Delay Model assumes a unit delay for all it's gates i.e., irrespective of it's type and load it's driving all the gates are assumed to have the same delay. This model while better than zero delay model is also inaccurate. This model can be represented probabilistically by taking into account higher order temporal and spatio-temporal dependencies.

4.1.3 Variable Delay Model

Variable Delay Model are of various types, this type of model, assigns different delays to gates based on their logic type, fanout, or the fanin of the gates they drive. In our work a fanout dependent delay has been assigned to gates, it is known that signals fall and rise slowly, if the number of fanout increases, owing to an increased load capacitance. In [[100]] Gate delay,d due to propagation has been represented as

$$d = f + p \tag{4.1}$$

where p represents parasitic delay, i.e., delay of a gate with no load capacitance, f represents effort delay, delay representing the size and fanout of the gate.Effort delay, f can be represented as

$$f = gh \tag{4.2}$$

here *g*, logical effort, represents the complexity of the gate, for example the logical effort of an inverter is said to be 1, while the logical effort of a 2-input NOR gate is said to be 5/3. fanout effort,*h*, is represented by

$$h = \frac{C_{out}}{C_{in}} \tag{4.3}$$

where C_{out} represents the external load being driven and C_{in} represents the input capacitance of the gate.

In this work, we consider the fanout effort, for delay assignment to individual gates. We neglect the input capacitance and approximate the load caapacitance of the gate as the number of fanout a gate has, this approximation is justified since the load capacitance increases linearly as the fanout increases.

4.2 Gate Delay Model

It has been proven that large fanouts cause the signals to fall and rise slowly, this is because gates with large fanouts would have to charge and discharge through large capacitances. In this work we have assigned variable fanout delay to the gates. S.Manich *et al* [9] have used the fanout delay model in their work to find the pair of vectors under which maximum weighted switching activity occurs.

Consider the [4.2.], gates have been assigned delay based on their fanouts. Gate $X4$, has been assigned a delay of 2 units, Gate $X5$ and $X6$ have been assigned a delay of 1 unit. From the assigned delays, Signal arrival times at inputs of gate have been calculated. Based on the signal arrival time and the delay of gate, time instants at which a possible signal transition can occur have been calculated and associated with the gates. The input signals have been assumed to be the output of a register. This assumption has enabled us to take into account just the previous input vector and present input vector for accounting undesired switching activity due to glitches. The previous input vector has been assumed to have taken place at time $-0t$, and the present input vector has been assumed to have taken place at time $0t$. In [4.2.], the input vectors {*X1,X2,X3*}, have been assigned two time instants {*-0t,0t*}.

Gate {*X4*}, has been assigned two time instants {*-0t,2t*}. {*X4(-0t)*} is affected by inputs {*X1(-0t),X2(-0t)*}, it has to be noted here that {*X4(-0t)*}, represents steady state output at {*X4*}, due to previous input vectors. {*X4(2t)*} is affected by inputs {*X1(0t),X2(0t)*}. Since only two time instants are associated

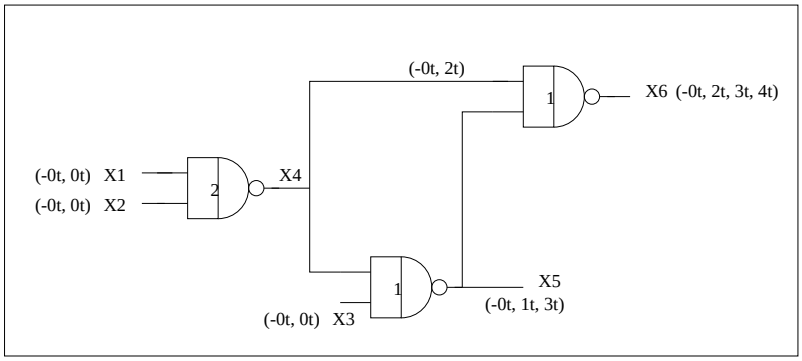


Figure 4.2. Variable Delay.

with $\{X4\}$, only one transition could occur at this node, due to transition from previous input vector to present input vector and the transition is represented by $\{X4(-0t) \rightarrow X4(2t)\}$.

Gate $\{X5\}$, has been assigned three time instants $\{-0t, 1t, 3t\}$. $\{X5(-0t)\}$ is affected by inputs $\{X4(-0t), X3(-0t)\}$. $\{X5(1t)\}$ is affected by inputs $\{X4(-0t), X3(0t)\}$. $\{X5(3t)\}$ is affected by inputs $\{X4(2t), X3(0t)\}$. Three time instants are associated with the node $\{X5\}$. The two possible transitions, associated with this node can be represented by $\{X5(-0t) \rightarrow X5(1t)\}, \{X5(1t) \rightarrow X5(3t)\}$.

Gate $\{X6\}$, has been assigned four time instants $\{-0t, 2t, 3t, 4t\}$. $\{X6(-0t)\}$ is affected by inputs $\{X4(-0t), X5(-0t)\}$. $\{X6(2t)\}$ is affected by inputs $\{X4(-0t), X5(1t)\}$. $\{X6(3t)\}$ is affected by inputs $\{X4(2t), X5(1t)\}$. $\{X6(4t)\}$ is affected by inputs $\{X4(2t), X5(3t)\}$. Three possible transitions associated with this node can be represented by
$\{X6(-0t) \rightarrow X6(2t)\}, \{X6(2t) \rightarrow X6(3t)\}\{X6(3t) \rightarrow X6(4t)\}$.

If the circuit had been represented as a zero-delay model, then we would have captured atmost 1 transition in the gates, and with unit-delay model, we would have captured a limited number of transitions in the gates, while in Variable delay model, we could capture all possible spurious transitions and also propagate these spurious transitions to the successive nodes.

4.3 Expanded Circuit Representation

The Circuit in [4.2.] has been expanded by creating instances of the gate, for all possible time instants at which a desired or an undesired, event could occur at the ouput of the gate. The gate *X4* would have 2 instances, *(X4,X4_1)*, corresponding to time instants $(-0t, 2t)$, associated with this gate. Gate

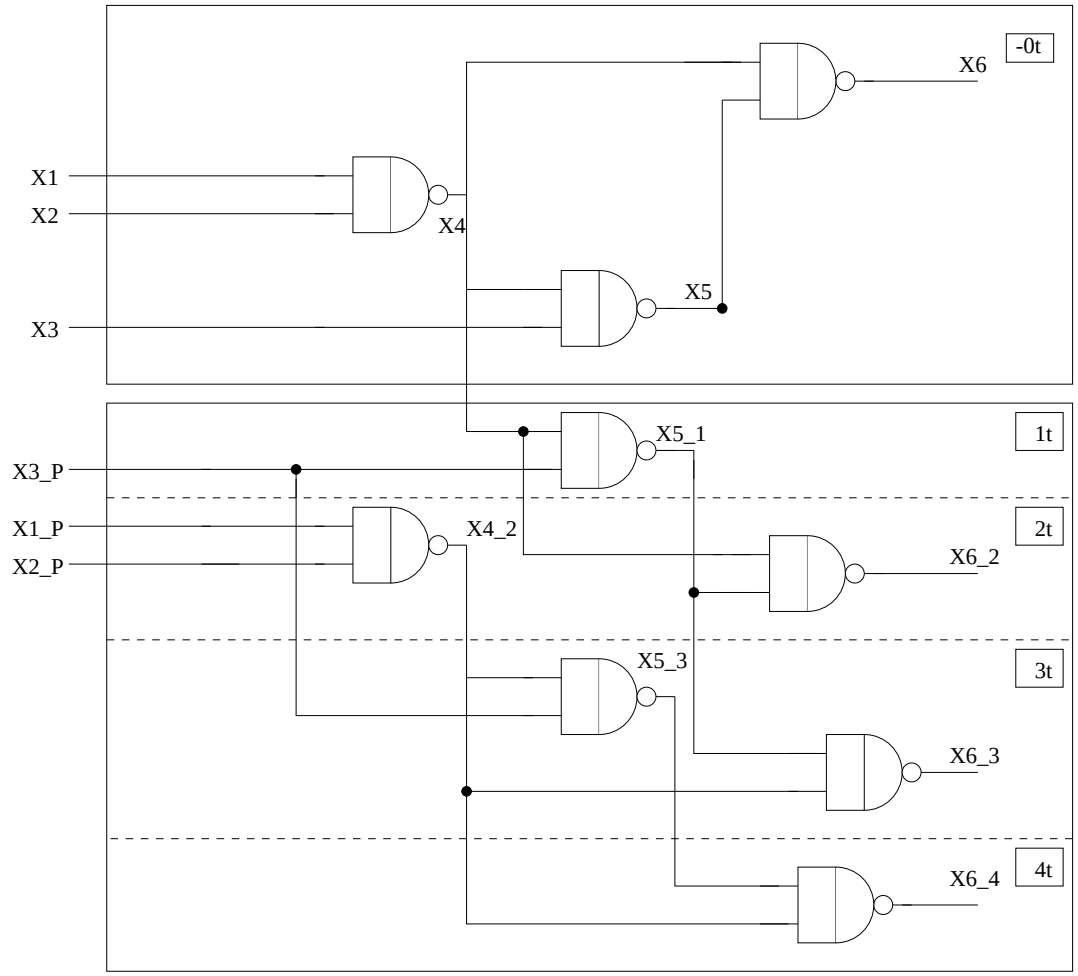


Figure 4.3. Expanded Circuit.

X5 would have 3 instances, *(X5,X5_1,X5_3)*, corresponding to time instants $(-0t, 1t, 3t)$ gate *X6* would have 4 instancess, *(X6,X6_2,X6_3,X6_4)*, corresponding to time instants $(-0t, 2t, 3t, 4t)$. The expanded figure [4.3.] has been created by forming 5 slots, corresponding to time instants, $(-0t, 1t, 2t, 3t, 4t)$. The various instances of gates are placed in the slots corresponding to their assigned time instant. The gates have been connected to their appropriate time instances of input signal. For example gate *X5_3*, would have at it's input, lines from gates *X3_1* and *X4_2* . Thus a connection is formed between all nodes.

4.4 Real Delay Probabilistical Model

In the last chapter, we had seen the construction of a Zero-Delay Bayesian model. It has to be noted here that the Zero-Delay Bayesian model captures higher order Spatial Correlations, but it captures *only* first order temporal and spatiotemporal correlation, zero-delay model would capture only 1 transition at a node for a given clock cycle and this would be achieved accurately, if we have the knowlege of node's previous and present state. In this work, we are representing a Real-Delay model of the circuit as Bayesian Network, therefore we would have to capture all the transitions that a node would experience. We successfully do that by modeling explicitly the higher order temporal correlations and higher order spatiotemporal correlations. The Real-Delay DAG structure is modelled on the expanded circuit,[4.4.]. The nodes in the DAG represent the gates at different time instances, the arcs represent the dependency. It has to be emphasized here that the Real-Delay DAG model differs from the Zero-Delay LIDAG-BN, in that the activity of a node is captured at various time instances a possible event could occur, owing to different input signal arrival times. In this Probabilistic model, all the nodes are assumed to have 4 states representing $\{(0 \to 0)(0 \to 1)(1 \to 0)(1 \to 1)\}$. The states represent the transition of node state X, $(X_{t-1} \to X_t)$, $X \in \{0,1\}$, from previous time instant, $t-1$ to present time instant,t. Probability values are assigned to these states, based on the conditional probability table assigned to the nodes, and the probability values of the parent signals. The conditional probability table reflects the nature of the gate. The conditional probability tables in this Real Delay Probabilistical Model, are identical to those of the LIDAG-BN, discussed in the last chapter. The construction of the Real Delay DAG, and the various dependencies modelled are discussed below,

The figure [4.4.] is a DAG structure for a Real Delay model.

4.4.1 Dependencies in Primary Inputs

Unlike the LIDAG-BN model, where we have assigned one set of input transition probabilities for all the primary inputs,in this model we have assigned two sets of input transition probability for primary inputs, one representing the anterior transition probabilities and the other representing the present transition probabilities. Let's say *t*, represents the time of present state, *t-1*, time of the previous state and *t-2*, represent the time of state before the previous state. In [4.4.], the anterior input tran-

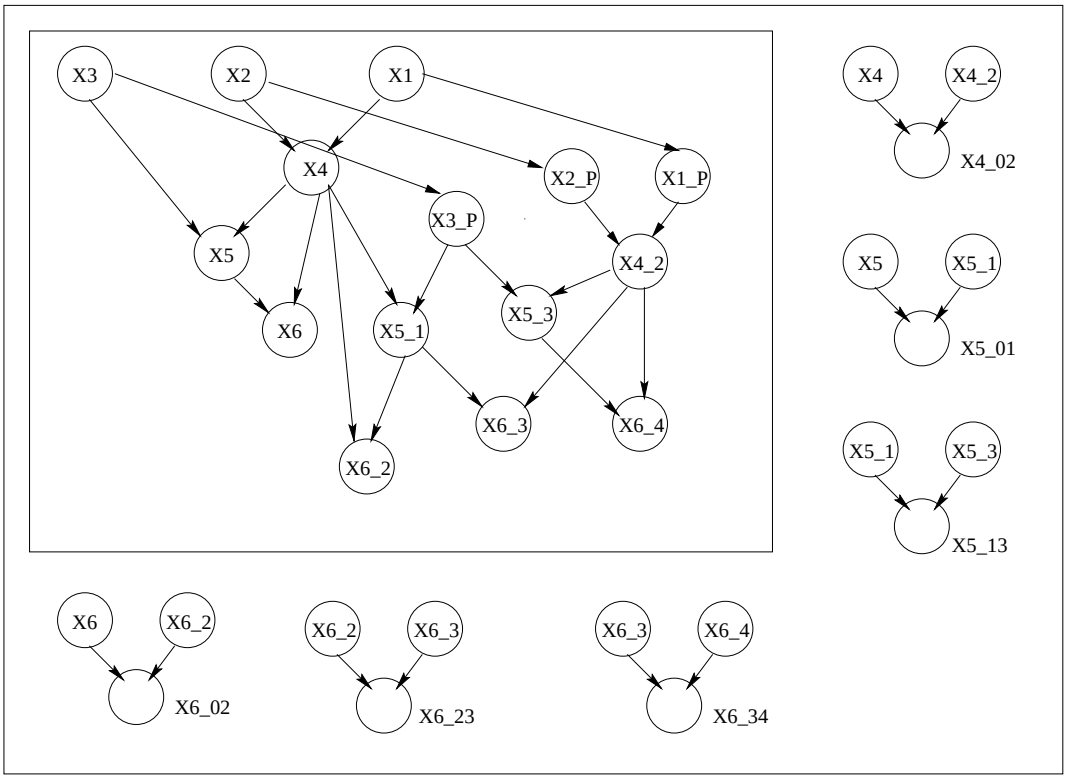


Figure 4.4. DAG of A Real-Delay Model.

Table 4.1. Conditional Probability Specifications for Primary Input Dependency.

Temporal Dependency of Primary Input				
$P(X1\_P \mid X1)$ for $X1\_P =$				$X1$
$\{x_{00}$	x_{01}	x_{10}	$x_{11}\}$	=
0.5	0.5	0	0	x_{00}
0	0	0.5	0.5	x_{01}
0.5	0.5	0	0	x_{10}
0	0	0.5	0.5	x_{11}

sition probability of primary input vectors {*X1,X2,X3*}, represents the switching probability, from time instance,$\{(t-2) \rightarrow (t-1)\}$ and the present input transition probability of input vectors {*X1_P,X2_P,X3_P*}, represents the switching probability, from time instance $\{(t-1) \rightarrow (t)\}$. It should be noted here that we are explicitly modeling the temporal correlation between the two switching transitions, by making the present transition depend on the previous transition. Since we are specifying the input vector states over three time instants,$\{(t-2),(t-1),(t)\}$, the input vectors can take 8 states namely $\{(000),(001),(010),(100),(101),($ The dependency of the node,*X1_P*, representing a primary input node with Present input transition probabilities, on *X1*, representing the primary input node with Previous input transition probabilities, is given by the following conditional proabability table.

4.4.2 Higher Order Spatio-Temporal Dependencies

In our Real-Delay DAG [4.4.], let's consider the node *X6_3*, it's states, symbolize the switching probabilities of gate, *X6*, from $\{(t-1)+3i\} \rightarrow \{t+3i\}$. The variable *i*, can be defined as a unit fan-out dependent delay, such that, if *N*, is the maximum time instant of a circuit, then $\{t\} < \{t+Ni\} < \{t+1\}$, here *t* and *t+1*, would represent the time instances at which present primary input vector and next primary input vector arrive from a register.

The parents of the node, *X6_3* are the nodes *X4_2*, and *X5_1*. The states of node*X4_2*, symbolize the switching probabilities of gate, *X4*, from $\{(t-1)+2i\} \rightarrow \{t+2i\}$. The states of node*X5_1*, symbolize the switching probabilities of gate, *X5*, from $\{(t-1)+1i\} \rightarrow \{t+1i\}$. The switching probability of node, *X6_3*, given the knowledge of nodes, *X4_2* and *X5_1* is represented as, $P(X6\_3 \mid X4\_2X5\_1)$, the

conditional probability quantifying this dependence is the same as that for Zero-Delay LIDAG BN, given in [3.1.]. This dependency denotes a *Higher Order Spatio-Temporal Dependency*, Since the nodes, $\{$ *X6_3, X4_2, X5_1*$\}$, are removed spatialy and temporally, by a higher order.

The gate *X6*, is denoted by the variables, $\{$*X6, X6_2, X6_3, X6_4*$\}$, at various time instants. The parents of these nodes are given by $\{$*(X4,X5),(X4,X5_1),(X4_2,X5_1),(X4_2,X5_3)*$\}$. Thus the Switching probability of gate X6, is updated at various time intervals. This updation, brings in a *Dynamic* nature to the Bayesian Network, enabling us to probabilistically, capture all the desired and undesired transitions, seen by the gate.

The Switching probabilities of a node can be calculated, if we have the knowledge of it's immediate parents, irrespective of the probabilities at other nodes. This has been made possible by the conditional independencies, portrayed by the DAG structure of Bayesian Networks.

4.4.3 Higher Order Temporal Dependencies

In [4.4.], the various instances of node X4 are $\{$*X4,X4_2*$\}$, here Switching probabilities associated with node *X4*, symbolize the switching of the states from $\{X4_{t-2} \rightarrow X4_{t-1}\}$, node *X4_2* symbolizes swiching probabilities from $\{X4\_2_{(t-1)+2i} \rightarrow X4\_2_{(t)+2i}\}$. To calculate the switching probability of node $X4$, from $\{X4_{t-1} \rightarrow X4_{t+2i}\}$, a child node *X4_02* is created, with $X4$ and $X4\_2$, as it's parents. The node,*X4_02* captures the total switching probability of gate, *X4* . The desired switching probability is the sum represented by $\{P(X4\_02_{t-1\rightarrow t+2i}(0 \rightarrow 1)\} + \{P(X4\_02_{t-1\rightarrow t+2i}(1 \rightarrow 0)\}$. The conditional probability of the node *X4_02*, is given in [4.2.].

Similarly the total switching probability of gate *X5*, is the sum of the $(0 \rightarrow 1) and (1 \rightarrow 0)$, switching probability of nodes, *X5_01* and *X5_13*. The parents of nodes $\{X5\_01_{t-1\rightarrow t+1i}, X5\_13_{t+1i\rightarrow t+3i}\}$, are the nodes $\{(X5, X5\_1), (X5\_1, X5\_3)\}$. The total switching probability of gate *X6* is the sum of $(0 \rightarrow 1) and (1 \rightarrow 0)$, switching probability of nodes, $\{X6\_02_{t-1\rightarrow t+2i}, X6\_23_{t+2i\rightarrow t+3i}, X6\_34_{t+3i\rightarrow t+4i}\}$. The parents of these nodes are $\{(X6, X6\_2), (X6\_2, X6\_3), (X6\_3, X6\_4)\}$.Let's consider a node X, whose time instants are defined in a set $Z = \{0,2,3...N\}$, The total switching probability at a node X is given by the equation,

$$P(X) = \sum_{i=1}^{N} P(X\_(i-1)(i)) \tag{4.4}$$

Table 4.2. Conditional Probability Specifications for Output Nodes.

Temporally connected 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}
1	0	0	0	x_{00}	x_{10}
0	1	0	0	x_{00}	x_{11}
0	0	1	0	x_{01}	x_{00}
0	0	0	1	x_{01}	x_{01}
0	0	1	0	x_{01}	x_{10}
0	0	0	1	x_{01}	x_{11}
1	0	0	0	x_{10}	x_{00}
0	1	0	0	x_{10}	x_{01}
1	0	0	0	x_{10}	x_{10}
0	1	0	0	x_{10}	x_{11}
0	0	1	0	x_{11}	x_{00}
0	0	0	1	x_{11}	x_{01}
0	0	1	0	x_{11}	x_{10}
0	0	0	1	x_{11}	x_{11}

given, $i, i-1 \in Z$

The dependency thus modelled, represents a *Higher Order Temporal Dependency,* between various time instances of a gate. The conditional probability of the nodes $\{(X4\_02),(X5\_01),(X5\_13),(X6\_02),(X6\_23),(X6\_34)\}$ is given in [4.2.].

The Switching probability thus calculated accurately captures the effect of Real Delay on a Gate, by explicitly modelling for the higher order temporal and higher order Spatiotemporal correlations, which the previous works have failed to account for.

In this chapter, we have seen, how to construct a Real-Delay model and discussed it's probabilistic nature. In the next chapter we would discuss some of the Inference Techniques used in Bayesian Networks.

CHAPTER 5

INFERENCE TECHNIQUES

In the Previous chapters, we had discussed the Graphical Representation of the VLSI circuits. The main purpose of this Graphical Representation has been to facilitate human reasoning and to reduce computational complexity of the probabilistic model. Numerical models of probability have relied on the joint probability density functions, to compute conditional probabilities, $P(x_i|x_j) = P(x_i,x_j)/P(x_j)$. Calculation of the joint probability involve complex computation. while Bayesian Networks constructed with conditional independency as its core concept represents x_j, as a frame of knowledge and defines x_i in the context of x_j i.e., $x_i|x_j$. The joint distribution function is calculated by rewriting the above formula. $P(x_i,x_j) = P(x_i|x_j).P(x_j)$.

The above representation called product rule, can be generalize to form a chain rule formula. [[40]] states that if we have a set of n events $E_1,E_2,...,E_n$, then the joint probability of these events can be written as, $P(E_1,E_2,...,E_n) = P(E_n|E_{n-1},....,E_2,E_1)....P(E_2|E_1)P(E_1)$.

Inversion Formula:

Bayesian inversion formula follows directly from the conditional probability formula represented above, $p(x_i|x_j = e) = P(x_j = e|x_i).P(x_i)/P(x_j = e)$

The above formula explains the belief updation of variable x_i, once x_j has been assigned an evidence, 'e'. $P(x_i)$ represents the 'prior' probability i.e., probability of $P(x_i)$, before x_j was assigned an evidence. $P(x_i|x_j = e)$, represents the 'posterior' probability i.e., the updated belief once an evidence has been assigned. $P(x_j = e|x_i)$ represents the 'likelihood' of obtaining the evidence given x_i and $P(x_j = e) = P(x_j = e|H).P(H) + P(x_j = e| - H).P(-H)$. The calculation of $P(x_j = e)$ is trivial for smaller networks, but as the network size increases, it's computation becomes complex. Different Inference algorithms exist to calculate $P(x_j = e)$. In this chapter we will discuss about some of the algorithms used for Bayesian Inference.

5.1 Exact Inference

5.1.1 Polytree Algorithm

Pearl suggested an efficient message propagation inference algorithm for polytrees, the algorithm was exact, but had a polynomial complexity in the number of nodes. This method could work only for singly connected networks. Loop cutset conditioning , was later introduced by pearl, to account for multiple connect networks. This method instantiates a selected subset of nodes, referred to as a loop cutset and changes the connectivity of a network into a single connected network. This single connected network, is solved using the polytree algorithm, he had suggested earlier. The complexity of this method grows exponentially with the size of loop cutset, making it hard to use on large networks.

5.1.2 Clustering

The clique-tree propagation algorithm suggested by Lauritzen and Spiegelhalter, also called the clustering algorithm, has been the most popular exact Bayesian Network algorithm. This method was used by Bhanja *et al.* in their work to estimate zero delay switching activity. In this method the directionality is removed from the DAG structure to form an undirected graph, called moral graph. This moral graph is triangulated by adding links to form cycles of 3 nodes. A collection of completely connected nodes is called clique, a junction tree of cliques is formed, with nodes representing cliques. Between any two cliques c_i, c_j in a tree there is a unique path, and elements in the intersection set $C_i \cap C_j$ are present in all the cliques in the path between C_i and C_j, this property, called running intersection property is used for local message passing based update algorithm. Steps involved in the propagation of evidence in the junction tree is given as a flowchart, taken from the work of Bhanja *et al.*

5.2 Approximate Sampling Algorithms

5.2.1 Stochastic Sampling Algorithms

Stochastic sampling algorithms are widely used approximate inference algorithms in Bayesian Networks. Sampling involves inferring some property of a large set of elements from the properties of a small, randomly selected, subset of elements. This method generates random samples according to

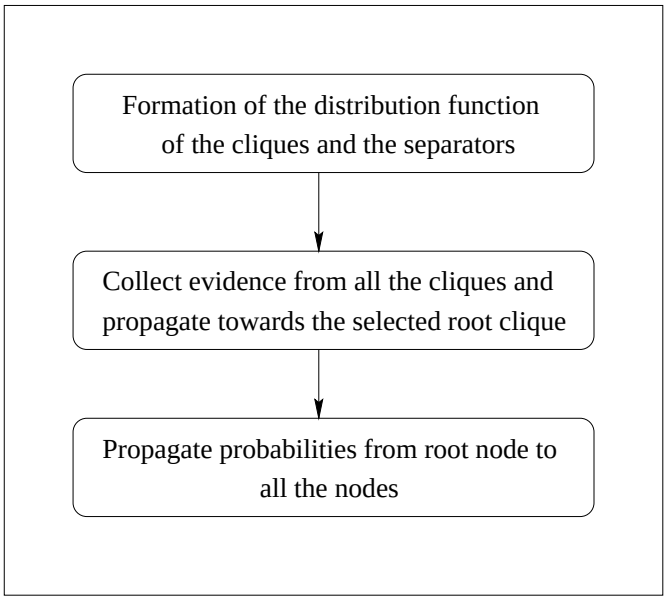


Figure 5.1. Flowchart.

the conditional property tables in the model, and estimate probabilities of query variables based on the frequency of their appearance in the sample.[[98]] . Accuracy of this method, is controlled by the number of samples irrespective of the structure of the network. Stochastic simulation algorithms can be divided into two main categories: importance sampling algorithms and Markov Chain Monte Carlo Methods(MCMC).We have used some of the following Stochastic sampling algorithms for inference.

5.2.2 Probabilistic Logic Sampling

Max Henrion[[99]] suggested Probabilistic Logic Sampling(PLS), a scheme employing stochastic simulation for probabilistic inference in large, multiply connected networks. In this approach a Bayesian Network is represented by a finite sample of deterministic scenarios generated at random from the probabilistic model. The accuracy of this method can be controlled by the sample size.

A random generator is used to produce a sample truth value for each source variable based on its probabilities, the evidence is propagated through the network in the direction of directed arrows, based on the conditional probability table of the nodes through which it traverses. This process is repeated for n times, where n is the sample size specified by the user. The prior marginal probability of a node is obtained as a fraction of times, the value of the node turns out to be true, with respect to the sample size.

The posterior probability for any event conditional on any set of observed variables can be estimated as the fraction of sample scenarios in which the event occurs out of the scenarios in which the condition occurs[[99]].

PLS is a forward sampling algorithm, i.e., the belief is propagated in only one direction. This method ignores samples which do not meet with the evidence values. In diagnostic inference, if a rare event is given as evidence, most of the samples would be discarded, thus severely hampering the accuracy. Also, increasing the number of evidences would severely affect accuracy. But, this method has proven to work well for predictive inferences.

5.2.3 Importance Sampling

Importance Function:

[[95]].Let f(X) be a function of n variables $X = (x_1, ..., x_n)$ over domain $\Omega \subset R^n$. Consider the problem of estimating the multiple integral

$$I = \int_{\Omega} f(X)dX \tag{5.1}$$

We assume that the domain of integration of f(x) is bounded, i.e., I exists. Importance sampling approaches this problem by estimating

$$I = \int_{\Omega} \frac{f(X)}{g(X)} g(X)dX \tag{5.2}$$

$g(x)$, called the importance function, is a probability density function such that $g(X) > 0$ for any $x \subset \Omega$. $g(X)$, should be easy to sample from. To estimate the integral, we generate samples $X_1, X_2, ..., X_N$ from $g(X)$ and use the generated values in the sample-mean formula.

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i}{g(X_i)} \tag{5.3}$$

Importance sampling assigns more weight to regions where $f(X) > g(X)$ and less weights were $f(X) < g(X)$ to correctly estimate I. If $f(X) > 0$, the optimal importance function is

$$g(X) = \frac{f(X)}{I} \tag{5.4}$$

However, since finding I is equivalent to solving the integral, a function close enough to the importance function is used to get good convergence rate. There are various algorithms that use a revised importance distribution for sampling as an approximation to the posterior distributions. Adaptive Importance Sampling and Evidence prepropogation Importance Sampling are such algorithms.

5.2.4 Adaptive Importance Sampling

Adaptive Importance Sampling [[97]]adresses the issue of inaccuracy during diagnostic inference in Probabilistic Logic Sampling, by learning a sampling distribution that is close to optimal importance sampling function. The DAG structure of the Bayesian Networks, is better suited for finding the Optimal importance function and belief updation. As new evidences appear the importance conditional property table(ICPT) of the nodes, similar in structure to conditonal property table, are updated. ICPT of a node X is a table of posterior probabilities $P(X|pa(X), E = e)$ conditional on the evidence and indexed by its immediate predecessors $pa(X)$. Importance function is constantly updated as new evidence arrives, a weighting function is introduced to the sampling results as the importance function moves further from the optimal importance function at different stages. Cheng performs two heuristic initializations that make this model better, 1. Initializing the ICPT table of the parents of the evidence nodes to uninform distribution improves convergence. 2. Setting a threshold probability value and replacing the prior probability of nodes that are less than the threshold value with the latter value. This method adopts an important function learning step to approach the optimal importance function, the learning step is a time consuming process. [Changhe] has introduced an algorithm that would directly compute an approximation of the optimal importance function, rather than learning it. We would discuss that algorithm in the next section.

5.2.5 Evidence Pre-propagation Importance Sampling Algorithm

EPIS algorithm [[95]] makes use of the loopy belief propagation [[40]] to calculate the importance function. In Loopy belief propagation, if E denotes a set of evidence, at a node X, E^+, would denote the evidence connected to X via it's parents and E^-, would denote the evidence connected to X via it' children. the posterior belief at X would be given by,

$$BEL(x) = \alpha P(e_x^- | x) P(x | e_x^+) \tag{5.5}$$

$$BEL(x) = \alpha \lambda(x) \pi(x) \tag{5.6}$$

where $\pi(x)$ denotes the message communicated by x's parent, and $\lambda(x)$ denotes the message communicated by x's child.

The following Theorem shows that Importance function can be calculated directly from poly trees.

THEOREM: 1[[95]] Let X_i be a variable in a polytree, and E be the set of evidence. The exact ICPT $P(X_i | pa(X_i, E)$ for X_i is

$$\alpha(pa(X_i)) P(X_i | pa(X_i)) \lambda(X_i) \tag{5.7}$$

where $\alpha(pa(X_i))$ is a normalizing constant dependent on $pa(X_i)$.

Proof: Refer to [[95]]

Corollary[[95]]: For a polytree, the optimal importance function is given by,

$$\rho(X|E) = \prod_{i=1}^{n} \alpha(pa(X_i)) P(X_i | pa(X_i)) \lambda(X_i) \tag{5.8}$$

using loopy belief propataion, thus an optimal importance function is obtained.

Similar to AIS, this method also uses some threshold, ε, for replacing smaller probability in the network by ε. The posterior probabilities are calculated from the samples that are generated based on the importance function.

CHAPTER 6

RESULTS

The ISCAS-85 Benchmark suite, introduced in netlist form at the Internation Symposium of Circuits and Systems in 1985, has been used in this work. The ISCAS-85 circuits have well-defined, high level structures and functions based on building blocks such as multiplexers, ALU's, and decoders. In this work, we have proposed a probabilistic switching model ,taking into account the effects of Gate delay, this has been done by creating different instances of a node at different time instants, as explained in Chapter 5. The ISCAS-85 netlist has been modified to capture the temporal dependencies and spatio-temporal dependencies between the nodes at different time instants. This modified netlist is used for constructing the Bayesian Network, and a Simulator. The simulator results serve as a ground truth, the results from Bayesian Network are compared with the result from the simulator, and the errors have been tabulated.

The Conversion of ISCAS-85 netlist into a Delay netlist was done using C programming Language, the maximum time instance of each circuit has been tabulated[6.1.]. From this Delay nelist file, the Simulator, for verification purposes, and the Probabilistic Dependency model were created.

The Probabilistic Bayesian Network model has been implanted with Dynamic tendencies, by explicit modelling for higher order spatio-temporal and higher order temporal correlations. The Bayesian

Table 6.1. Maximum Time Instants for Some Combinational Benchmark Circuits.

Circuits	Maximum Time Instance
C432	55
C499	30
C880	51
C1355	55
C2670	81
C5315	90

Table 6.2. Switching Activity Estimation Error Statistics Based on Delay DAG Modeling, Using PLS Inference Scheme, Using 1000 Samples, for ISCAS'85 Benchmark Combinational Circuits.

Circuits	Average Error (E_μ)	Time (s) s ($CPU + I/O$)
C432	0.023689	4.501
C499	0.013762	2.734
C880	0.018867	17.422
C1355	0.018578	43.643
C2670	0.021974	93.59
C5315	0.034694	641.51

Network was created using GeNIe, a tool from Decision System Laboratories, University of Pittsburgh, the inferencing was done using Probabilistic Logic Sampling, an efficient stochastic inference method available in the same tool. The tests were performed using Pentium IV, 2.00GHz, Windows XP computer.

The Simulator requires two input vectors, the previous input vector and the present input vector, at any given time, to account for spurious transition due to delay. The vectors were made dependent temporally. The Simulation was run for 100000 vectors, the vectors were generated using a pseudo-random generator and the switching value of each node was calculated.

The Bayesian Networks requires the probability of two Switching input transitions, the anterior input transition and present input transition. The results were obtained by basing the input switching probabilities, for all the four states,$\{$00,01,10,11$\}$ of all primary input nodes as 0.25. The temporal dependencies between the primary inputs was modelled explicitly in the Bayesian Network, as were the higher order temporal and higher order spatio-temporal correlations among the nodes. The Inferencing was done using Probabilistic Logic Sampling, for a sample size of 1000. The results when compared to the Simulation results have showed very low mean error, evident in the given table[6.2.].

The table listed shows the average error E_μ, between the results from the probabilistic model and the simulation model. The table also highlights the time efficiency of our probabilistic model. The elapsed time was obtained from the WINDOWS environment and it is the sum of the CPU, memory access and I/0 time.

This Probabilistic model as evident from the results has a low computation time and a high accuracy. This high accuracy is attributed to the capturing of the input dependencies, and higher order spatial, higher order temporal and higher order spatio-temporal correlations in the circuit.

CHAPTER 7

CONCLUSION

Power Dissipation has been a major concern for researchers in the VLSI Design field. It is known that Switching Activity at gates, has been a major contributor to Power Dissipation. Switching Activity Estimation has been carried out at all levels of design abstraction, and it has been proven that Gate Level switching estimation has a better accuracy compared to the other levels.

The Literature Review shows that switching estimation has been carried out using Simulation , Statistical simulation and Probabilistic methodologies. The Simulation and Statistical simulation based methodologies have proven to be highly input pattern sensitive. To acquire an accurate result using these methodologies, a large input set would be required, increasing the time complexity of the process. The existing Probabilistic methods do not have a unified framework that accounts for the different issues, like input pattern dependency, spatial correlations, temporal correlations and spatio-temporal correlations, in the Switching Estimation process, Thus failing to accurately capture the Swtiching Activity.

The work of Bhanja *et al*, while accounting for higher order spatial correlations and first order temporal correlation, fail to account for Gate Delays, ignoring the spurious transitions that occur due to different arrival times of the input signals. The other Probabilistic models that account for delay, do not model higher order temporal and higher order spatio-temporal correlations. The Gate Delay model in this work, uses Bayesian Networks to accurately capture the spurious transitions, by explicitly modelling for higher order spatio-temporal and higher order temporal correlations. The inferencing of results has been done using Probabilistic Logic sampling, a proven efficient inferencing mechanism for predictive Probabilistic inference. The results when compared to the simulation results have minimum error and the computation time is very low, compared to other methods.

7.1 Future Work

1. In the Nano domain, Interconnect Delay has been gaining dominance. This work can be extended to account for interconnect delay, by altering the delay assignment mechanism.

2. In this work we have effectively developed a Probabilistic Switching model for Combinational circuits, under Real Delay conditions. We could extend this work for switching estimation in sequential circuits.

REFERENCES

[1] S. Ramani, "Graphical Probabilistic Switching Model : Inference and Characterization of Power Dissipation in VLSI Circuits", *Master's Thesis.*

[2] N. Ramalingam, "A Complete Probabilistic Framework for Learning Input Models for Power and Crosstalk Estimation in VLSI Circuits", *Master's Thesis.*

[3] S. Bhanja, and N. Ranganathan, "Dependency Preserving Probabilistic Modeling of Switching Activity using Bayesian Networks", *Proc. of 38th IEEE/ACM Design Automation Conference (DAC)*, pp. 209–214, 2001.

[4] 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.

[5] S. Bhanja, and N. Ranganathan,"Switching Activity Estimation of Large Circuits using Multiple Bayesian Networks", *Proc. of 7th Asia and South Pacific Design Automation Conference and 15th International conference on VLSI Design*, pp. 187–192, 2002.

[6] S. Bhanja, and N. Ranganathan,"Modeling Switching Activity Using Cascaded Bayesian Networks for Correlated Input Streams", To appear in *Intl. Conference on Computer Design*, 2002.

[7] 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.

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

[9] S. Manich and J. Figueras, "Maximizing the weighted switching activity in combination CMOS circuits under the variable delay model" *Proceedings of European Design and Test Conference.*, pp. 597-602. 1997.

[10] 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.

[11] D. Kim, and T. Ambler,"Robust transition density estimation by considering input/output transition behavior", *International Symposium on Circuits and Systems*, vol. 5, pp. 403–406, 2001.

[12] R. Marculescu, D. Marculescu, and M. Pedram, "Probabilistic Modeling of Dependencies During Switching Activity Analysis", *revised version submitted to IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*,

URL http://atrak.usc.edu/~massoud/sign_download.cgi?pecp-journal.ps
Access date August 22, 2002.

[13] 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, no. 2, pp. 73–83, February 1998.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] C.-Y. Tsui, M. Pedram and A. M. Despain,“Efficient Estimation of Dynamic Power Consumption under a Real Delay Model”, *Proceedings of International Conference on Computer Aided Design*, pp. 224–228, 1993.

[29] 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.

[30] 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.

[31] J. Monteiro, S. Devadas, A. Ghosh, K. Keutzer, and J. White,”Estimation of average switching activity in combinational logic circuits using symbolic simulation”,*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 1, pp. 121–127, 1997.

[32] J. C. Costa, J. C. Monteiro, and S. Devadas, “Switching activity estimation using limited depth reconvergent path analysis”, *International Symposium on Low Power Electronics and Design*, pp. 184–189, 1997.

[33] T.-L. Chou, K. Roy, and S. Prasad, “Estimation of circuit activity considering signal correlations and simultaneous switching”, *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 300–303, 1994.

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

[35] W. Lam, and F. Bacchus, “Using causal information and local measures to learn bayesian networks,” in *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pp. 243–250, 1993.

[36] `URL http://www.hugin.com/.`

[37] D. Heckerman, D. Geiger, and D. Chickering, “Learning bayesian networks: The combination of knowledge and statistical data,” Tech. Rep. MSR-TR-94-09, Microsoft Research, 1994.

[38] S. Kullback, and R. A. Leibler, “ Information and Sufficiency”, *Ann. Math. Statistics*, vol. 22, pp. 79–86, 1951.

[39] R. Dechter, “Topological Parameters for Time-space Tradeoffs”, *Uncertainty in Artificial Intelligence*, pp. 220–227, 1996.

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

[41] C. K. Chow, and C. N. Liu, "Approximating Discrete Probability Distributions with Dependence Trees", *IEEE Transaction on Info. Theory*, vol. 14, pp. 462–467, 1968.

[42] Y. Xiang , K. G. Olesen, and F. V. Jensen, "Practical issues in modelling diagnostic systems with multiply sectioned Bayesian networks", *International Journal of pattern Recognition and Artificial Intelligence*, vol. 14, no. 1, pp. 59–71, 2000.

[43] G. F. Cooper, "The Computational Complexity of Probability Inference Using Bayesian Belief Networks",*Artificial Intelligence*, vol. 42, pp. 393–405, 1990.

[44] M. Buhler, M. Papesch, K. Kapp, and U. G. Baitinger,"Efficient Switching Activity Simulation Under a Real Delay Model Using a Bit-Parallel Approach", *Proc. of the European Design Automation and Test Conference*, pp. 459–463, 1999.

[45] 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.

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

[47] 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.

[48] 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.

[49] D. Rabe, G. Jochens, L. Kruse, and Nebel W. Power- Simulation of Cell based ASICs: Accuracyand Performance Trade-Offs. In *Proc. of the European Design Automation and Test Conf.*, pp. 356–361, 1998.

[50] Rabe D., Nebel W. A New Approach in Gate-Level Glitch Modeling. In *Proc. of Design Automation and Test Conf. with EURO-VHDL*, pp. 66–71, 1996.

[51] 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.

[52] 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.

[53] 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.

[54] A. Macii, E. Macii, M. Poncino, and R. Scarsi,"Stream synthesis for efficient power simulation based on spectral transforms", *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 3, pp. 417–426, 2001.

[55] E. Macii, M. Pedram, and F. Somenzi, "High-level power modeling, estimation, and optimization", *IEEE Trans. Computer-Aided Design Inte-grated Circuits Syst.*, vol. 17, pp. 1061–1079, November, 1998.

[56] 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.

[57] 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.

[58] 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.

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

[60] 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.

[61] J. Kozhaya, and F. Najm, "Accurate power estimation for large sequential circuits", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp. 488–493, 2002.

[62] T-L. Chou, and K. Roy, "Statistical estimation of combinational and sequential CMOS digital circuit activity considering uncertainty of gate delays", *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 95–100, 1997.

[63] N. Zhu, R-D. Zhou, and X-Z. Yang, "A new approach on power estimation of CMOS sequential logic circuits", *Proceedings of 5th International Conference on Solid-State and Integrated Circuit Technology*, pp. 488–491, 1998.

[64] L-P. Yuan, C-C. Teng, and S-M. Kang, "Statistical Estimation Of Average Power Dissipation In Sequential Circuits", *The 34th Proceedings of Design Automation Conference*, pp. 377–382, 1997.

[65] C.-Y. Tsui, M. Pedram and A. M. Despain,"Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs",*Proceedings of the 31st Design Automation Conference*, pp. 18–23, June, 1994.

[66] C.-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain, and B. Lin, "Power Estimation in Sequential Logic Circuits", *IEEE Transactions on VLSI Systems*, pp. 404–416, 1996.

[67] J. Monteiro, S. Devadas, and A. Ghosh,"Estimation of Switching Activity in Sequential Logic Circuits with Applications to Synthesis for Low Power",*Proceedings of the 31st Design Automation Conference*, pp. 12–17, June, 1994.

[68] G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, "Probabilistic Analysis of Large Finite State Machines", *Proceedings of the 31st Design Automation Conference*, pp. 270–275, June, 1994.

[69] R. Burch, F. N. Najm, and T. Trick, "A Monte Carlo Approach for Power Estimation", *IEEE Transactions on VLSI Systems*, vol. 1, no. 1, pp. 63–71, March 1993.

[70] 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.

[71] A. Murugavel, N. Ranganathan, R. Chandramouli, and S. Chavali, "Average Power in Digital CMOS Circuits Using Least Square Estimation", to appear in *Proc. of Intl. Conf. on VLSI Design*, Jan 2001; also under review in IEEE Transactions on VLSI Systems.

[72] L. P. Yuan, C. C. Teng, and S. M. Kang, "Statistical estimation of average power dissipation in CMOS VLSI circuits using non-parametric techniques," *IEEE/ACM Intl. Symp. on Low Power Electronics and Design*, pp. 73–78, 1996.

[73] Y. J. Lim, K. I. Son, H. J. Park, and M. Soma, "A statistical approach to the estimation of delay dependent switching activities in CMOS combinational circuits," *Proceedings - ACM/IEEE Design Automation Conference*, June 1996.

[74] V. Saxena, F.N. Najm, and I.N. Hajj, "Monte-Carlo approach for power estimation in sequential circuits", *Proceedings of the European Design and Test Conference*, pp. 416–420, 1997.

[75] R. Chandramouli, and V.K. Srikantam, "On mixture density and maximum likelihood power estimation via expectation-maximization", *Asia South Pacific Design Automation Conference*, January 2000.

[76] S. Bilavarn, G. Gogniat, and J. L. Philippe, " Area time power estimation for FPGA based designs at a behavioral level Bilavarn", *7th IEEE International Conference on Electronics, Circuits and Systems*, vol. 1, pp. 524–527, 2000.

[77] F. Ferrandi, F. Fummi, E. Macii, M. Poncino, and D. Sciuto, "Power estimation of behavioral descriptions", *Proceedings of Design, Automation and Test in Europe*, pp. 762–766, 1998.

[78] P. Landman, R. Mehra, and J. M. Rabaey, "An integrated CAD environment for low-power design", *IEEE Design & Test of Computers* , vol. 13, no. 2, pp. 72–82, 1996.

[79] P. Landman, and J. M. Rabaey, "Activity-sensitive architectural power analysis", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 6, pp. 571–587, 1996.

[80] N. Kumar, S. Katkoori, L. Rader, and R. Vemuri, "Profile-Driven Behavioral Synthesis for Low Power VLSI Systems", IEEE Design & Test of Computers, Fall Issue, pp. 70–84, 1995.

[81] R. Vemuri, S. Katkoori, M. Kaul, and J. Roy, "An Efficient Hierarchical Register Optimization Algorithm for High Level Synthesis from Behavioral Specifications," ACM Transactions on Design Automation of Electronic Systems (TODAES) , vol. 7, no. 1, January 2002.

[82] D. Bruni, G. Olivieri, A. Bogliolo, and L. Benini, "Delay-sensitive power estimation at the register-transfer level", *IEEE International Conference on Electronics, Circuits and Systems*, vol. 2 , pp. 1031–1034, 2001.

[83] N. R. Potlapally, A. Raghunathan, G. Lakshminarayana, M. S. Hsiao, and S. T. Chakradhar, "Accurate power macro-modeling techniques for complex RTL circuits", *Fourteenth International Conference on VLSI Design*, pp. 235–241, 2001.

[84] S. Gupta, and F. N. Najm,“Analytical models for RTL power estimation of combinational and sequential circuits”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 7, pp. 808–814, July 2000.

[85] V. Krishna, R. Chandramouli, and N. Ranganathan, “Computation of lower bounds for switching activity using decision theory”,*IEEE Trans. on VLSI Systems*, vol. 7, no. 1, pp. 125–129, March 1999.

[86] M. Nemani, and F. N. Najm,”High-level area prediction for power estimation”,*Proc. of IEEE Custom Integrated Circuits Conference*, pp. 483–486, 1997.

[87] `http://public.itrs.net/Files/2000UpdateFinal/2kUdFinal.htm`.

[88] J. M. Rabaey, and M. Pedram, “Low Power Design Methodologies”, Kluwer Academic Publishers, 1996.

[89] A. Raghunathan, and Niraj Jha, “ High Level Power Analysis and Optimization”, Kluwer Academic Publishers, 1998.

[90] K. Roy, and S. C. Prasad, “ Low Power CMOS VLSI Circuit Design”, John Wiley & Sons Inc., 2000.

[91] Neil H.E. Weste, David Harris, ”CMOS VLSI DESIGN A Circuits and Systems Perspective”, Pearson Education Inc., 2005.

[92] J. Frenkil,”Tools and Methodologies for Low Power Design”, *Tutorial in ACM/IEEE Design Automation Conference*, 1997.

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

[94] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel and F. Baez, ”Reducing Power in High-performance Microprocessors”,*Proceedings - ACM/IEEE Design Automation Conference*, pp. 732–737, 1998.

[95] 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.

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

[97] J. Cheng and M.J. Druzdzel. AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian Networks. *Journal of Artificial Intelligence Research*,13:155–188, 2001.

[98] Haipeng Guo, William Hsu, ”A Survey of Algorithms for Real-Time Bayesian Network Inference”.

[99] 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.

[100] Neil H.E. Weste, David Harris, ”CMOS VLSI DESIGN A Circuits and Systems Perspective”, Pearson Education Inc.