

Estimation of Switching Activity in Sequential Circuits Using Dynamic Bayesian Networks

Sanjukta Bhanja*, Karthikeyan Lingasubramanian* and N. Ranganathan⁺

* Department of Electrical Engineering

⁺ Department of Computer Science and Engineering
University of South Florida, Tampa, FL

Abstract

We propose a novel, non-simulative, probabilistic model for switching activity in sequential circuits, capturing both spatio-temporal correlations at internal nodes and higher order temporal correlations due to feedback. This model, which we refer to as the temporal dependency model (TDM), can be constructed from the logic structure and is shown to be a dynamic Bayesian Network. Dynamic Bayesian Networks are extremely powerful in modeling high order temporal as well as spatial correlations; it is an exact model for the underlying conditional independencies. The attractive feature of this graphical representation of the joint probability function is that not only does it make the dependency relationships amongst the nodes explicit but it also serves as a computational mechanism for probabilistic inference. We report average errors in switching probability of 0.006, with errors tightly distributed around the mean error values, on ISCAS'89 benchmark circuits involving up to 10000 signals.

1 Introduction

The ability to form accurate estimates of power usage, both dynamic and static, of VLSI circuits is an important issue for rapid design-space exploration. Switching activity is one important component in dynamic power dissipation that is independent of the technology of the implementation of the VLSI circuit. Contribution to total power due to switching is dependent on the logic of the circuit and the inputs and will be present even if sizes of circuits reduce to nano domain. Apart from contributing to power, switching in circuits is also important from reliability point of view and hence can be considered to be fundamental in capturing the dynamic aspects of VLSI circuits.

Among different types of VLSI circuits, switching in sequential circuits, which also happens to be the most

common type of logic, are the hardest to estimate. This is particularly due to the complex higher order dependencies in the switching profile, induced by the spatio-temporal components of the main circuit but mainly caused by the state feedbacks that are present. These state feedbacks are not present in pure combinational circuits. One important aspect of switching dependencies in sequential circuits that one can exploit is the first order Markov property, i.e. the system state is independent of all past states given just the previous state. This is true because the dependencies are ultimately created due to logic and re-convergence, using just the current and last values.

The complexity of switching in sequential circuits arise due to the presence of feedback in basic components such as flip-flops and latches. The inputs to a sequential circuit are not only the primary inputs but also these feedback signals. The feedback lines can be looked upon as determining the state of the circuits at each time instant. The state probabilities affect the state feedback line probabilities that, in turn, affect the switching probabilities in the entire circuit. Thus, formally, given a set of inputs i_t at a clock pulse and present states s_t , the next state signal s_{t+1} is uniquely determined as a function of i_t and s_t . At the next clock pulse, we have a new set of inputs i_{t+1} along with state s_{t+1} as an input to the circuit to obtain the next state signal s_{t+2} , and so on. Hence, the statistics of *both* spatial and temporal correlations at the state lines are of great interest. It is important to be able to model both kinds of dependencies in these lines.

Previous work [4] has shown that a combinational circuit can be *exactly* modeled in a probabilistic Bayesian Network. The BN structure, however, cannot model cyclical logical structure, like those induced by the feedback lines. This cyclic dependence effects the state line probabilities that, in turn, effect the switching probabilities in the entire circuit.

In this work, we propose a probabilistic, non-

simulative, predictive model of the switching in sequential circuits using temporal dependency model (TDM) structure that explicitly models the higher order temporal and spatial dependencies among the feedback lines. The nodes in TDM represent switching random variable at the primary input, state feedback, and internal lines. These random variables defined over four states, representing four possible signal transitions at each line which are $(x_{00}, x_{01}, x_{10}, x_{11})$. Edges of TDM denote direct dependency. Some of them are dependencies within one time slice and the conditional probability specification capture the conditional probability of switching at an output line of a gate given the switching at the input lines of that gate. Rest of the edges are temporal, i.e. the edges are between nodes from different time slices, capturing the state dependencies between two consecutive time slices. We add another set of temporal edges between the same input line at two consecutive slices, capturing the implicit spatio-temporal dependencies in the input switchings. Temporal edges between just consecutive slices are sufficient because of the first order Markov property of the underlying logic. We prove that the TDM structure is a *Dynamic Bayesian Networks* (DBN) capturing all spatial and higher order temporal dependencies among the switchings in a sequential circuit. It is a minimal representation, exploiting all the independencies. The model, in essence, builds a factored representation of the joint probability distribution of the switchings at all the lines in the circuit.

For large circuits, we resort to a stochastic simulation based inference method which is non-simulative and is different from samplings that are commonly used in circuit simulations. In the later, the input space is sampled, whereas in our case both the input and the line state spaces are sampled simultaneously, using a strong correlative model, as captured by the Bayesian network. Due to this, convergence is faster and the inference strategy is input pattern insensitive.

The key features of our approaches are (1) this is the first and only probabilistic modeling framework that can be used to model switching in *both* sequential and combinational circuits and (2) the adopted strategy uses a dynamic Bayesian Network formalism which is edge-minimal, pattern insensitive switching model that are not logic unraveling[9].

2 Prior Work

Existing techniques for switching estimation in sequential circuits use statistical simulation. We are the first ones to use entirely probabilistic models for switching estimation in sequential circuits. Almost all the

statistical techniques in some way or the other employ sequential sampling of inputs, along with stopping criteria determined by the assumed statistical model. Stamoulis *et al.* [10] considered a path oriented transition probability computation to sample the input signal space, but the model did not account for correlations between latch and the combinational part. In one of the pioneering work, Najm *et al.* [7] proposed a logic simulation to obtain the state line probability and then used statistical simulation for estimation.

Tsui *et al.* [9] used a two-step approach. First, the stationary state line probabilities were estimated instead of state probabilities, thereby reducing the problem complexity but sacrificing the ability to model the spatial dependencies among the state lines. A set of non-linear equations described the relations of the next state to the primary inputs and the current state lines. These equations are solved using the Picard-Peano and Newton-Raphson methods to arrive at locally optimal solution for the individual state line probabilities. Next, given these state probability estimates, statistical simulations (or, for very small circuits, binary decision diagrams (BDD)) were used to obtain switching estimates. Yuan *et al.* [11] exploited the observation that beyond a length of samples the inputs can be considered independent. Saxena *et al.* [14] simulated multiple copies of a circuit, with mutually independent input vectors, thereby generating mutually independent samples. The entire sequential circuit estimates were formed by Monte-Carlo simulation with these inputs. Chen *et al.* [12] presented a technique to estimate upper and lower bounds of power by considering the signal probability and signal activity at the inputs. Kozhaya *et al.* [13] enhanced this method in their work to estimate power in sequential circuits.

Even the best simulation based methods suffer from weak input pattern dependencies. Besides, for any change in primary input statistics, simulations need to be rerun. For these reasons, we prefer probabilistic strategies, particularly those that model the underlying joint probability distribution of the node switching. Such probabilistic models, not only allows one to estimate the switching probabilities, but also readily facilitates conditional estimations, i.e. estimation conditioned on the knowledge of the switching statistics at some nodes, not necessarily the input lines. In this work, we propose a graphical probabilistic model over all the state nodes, inputs, and internal lines to accurately model the joint switching probabilities of the whole circuit. The model accounts for high order temporal and spatial dependencies. The exact spatio-temporal correlation is modeled over n time slices to capture higher (> 1) order temporal effects that are

present in sequential circuits. Circuits modeled using n slices captures n -th order temporal dependencies. Similar strategy was used by Tsui *et al.* [9], who “unraveled” the circuits n times to arrive at the statistics of the individual state lines, which were then used in simulations. We do not restrict ourselves to just state lines. Our comprehensive probability model, spanning multiple time slices, is over all the lines and simultaneously estimates the switching probabilities at all the lines (primary input, internal, and state lines).

3 Background on Dynamic Bayesian Networks

In this section, we discuss the structure and fundamentals of dynamic Bayesian Networks, which underlies our modeling of sequential circuits as TDMs. Since Dynamic Bayesian Networks are structurally Bayesian Networks themselves, we will highlight important features of Bayesian Networks as well. The following discussion is fashioned after [3, 2]. As a peek ahead, the reader can look at Fig. 3 for some examples of the representation for a small circuit.

A Bayesian network is a directed acyclic graph (DAG) representation of the conditional factoring of a joint probability distribution. Any probability function $P(x_1, \dots, x_n)$ can be written as¹

$$\begin{aligned}
 P(x_1, \dots, x_N) &= P(x_n | x_{n-1}, x_{n-2}, \dots, x_1) \\
 &\quad P(x_{n-1} | x_{n-2}, x_{n-3}, \dots, x_1) \\
 &\quad \dots P(x_1)
 \end{aligned} \tag{1}$$

This expression holds for any ordering of the random variables. In most applications, a variable is usually not dependent on all other variables. There are lots of conditional independencies embedded among the random variables, which can be used to reorder the random variables and to simplify the conditional probabilities.

$$P(x_1, \dots, x_N) = \prod_v P(x_v | Pa(X_v)) \tag{2}$$

where $Pa(X_v)$ are the parents of the variable x_v , representing its direct causes. This factoring of the joint probability function can be represented as a directed acyclic graph (DAG), with nodes (V) representing the random variables and directed links (E) from the parents to the children, denoting direct dependencies.

The DAG structure preserves all the *independencies* among sets of random variables and is referred to as a Bayesian network. The concept of Bayesian network can be precisely stated by defining the notion of *conditional independence* among a set of random variables.

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

3.1 DBN Structure

A Dynamic Bayesian Network (DBN) is a generalization of Bayesian networks to handle temporal effects of an evolving set of random variables. Other formalisms such as hidden Markov models and linear dynamic systems are special cases. The nodes and the links of the DBN are defined as follows. For any time period or slice, t_i , let a directed acyclic graph (DAG), $G_{t_i} = (V_{t_i}, E_{t_i})$, represent the underlying dependency graphical model for the combinational part. Then the nodes of the DBN, V , is the union of all the nodes each time slice.

$$V = \bigcup_{i=1}^n V_{t_i} \tag{3}$$

However, the links, E , of the DBN are not just the union of the links in the time-slice DAGs, but also include links between time-slices, i.e. temporal edges, $E_{t_i, t_{i+1}}$, defined as

$$E_{t_i, t_{i+1}} = \{(X_{i, t_i}, X_{j, t_{i+1}}) | X_{i, t_i} \in V_{t_i}, X_{j, t_{i+1}} \in V_{t_{i+1}}\} \tag{4}$$

where X_{j, t_k} is the j -th node of the DAG for time slice t_k . Thus the complete set of edges E is

$$E = E_{t_1} \cup \bigcup_{i=2}^n (E(t_i) + E_{t_{i-1}, t_i}) \tag{5}$$

Apart from the independencies among the variables from one time slices, we also have the following independence map over variable across time slices if we assume that the random variables representing the nodes follow Markov property, which is true for switching.

$$I(\{X_{j, t_1}, \dots, X_{j, t_{i-1}}\}, X_{j, t_i}, \{X_{j, t_{i+1}}, \dots, X_{i, t_{i+k}}\}) \quad \forall i > 1, k > 1 \tag{6}$$

4 TDM Modeling

The core idea is to express the switching activity of a circuit as a joint probability function, which can be mapped one-to-one onto a Bayesian Network, while preserving the dependencies. To model switching at a line, we use a random variable, X , with four possible states indicating the transitions from $\{x_{00}, x_{01}, x_{10}, x_{11}\}$. For combinational circuits, directed edges are drawn from the random variables representing switching of each gate input to the random variable for switching at the outputs of that gate. At each node, we also have conditional probabilities, given the states of parent nodes. If the DAG structure follows the logic structure then it is guaranteed to map all the dependencies inherent in the *combinational* circuit.

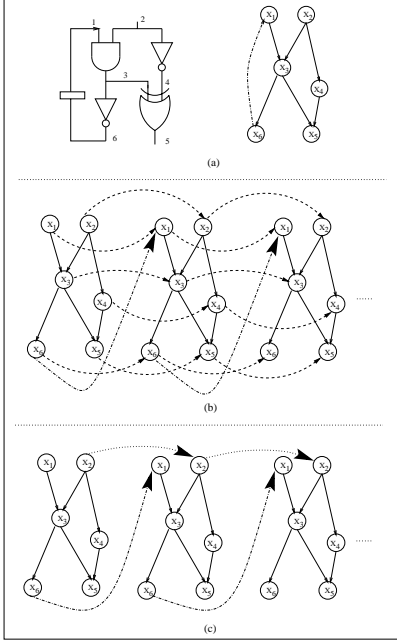


Figure 1. (a) Combinational circuit and its graph structure. (b) Time unraveled representation. (c) TDM representation

However, sequential circuits cannot be handled in this manner.

4.1 Structure

Let us consider graph structure of a small sequential circuit shown in Fig. 1(a). Following logic structure will not result in a DAG; there will be directed cycles due to feedback lines. To handle this, we do not represent the switching at a line as a single random variable, X_k , but rather as a set of random variables, representing the switching at consecutive time instants, $\{X_{k,t_1}, \dots, X_{k,t_n}\}$, and then model the logical dependencies between them by two types of directed links.

1. For any time instant, edges are constructed between nodes that are logically connected in the combinational part of the circuit, i.e. without the feedback component. Edges are drawn from each random variable representing switching activity at each input of a gate to the random variable representing output switching of the gate.
2. We connect random variables representing the same state line from two consecutive time instants, $X_{k,t_i}^s \rightarrow X_{k,t_{i+1}}^s$, to capture the temporal dependencies between the switchings at state lines.

Moreover, we also connect the random variables representing the switching at primary input lines at consecutive times, $X_{k,t_i}^p \rightarrow X_{k,t_{i+1}}^p$. This is done to capture the constraint in the primary line switching between two consecutive time instants. For instance, if an input has switched from $0 \rightarrow 1$ at time t_i , then switching at the next time instant cannot be $0 \rightarrow 0$.

Let us consider Figure 1(b), note that we have all random variables connected to the previous time slice for every slice. This is required as our random variables are switching and not logic. Every random variable in this experiment has effect in time stamps. However, Markov blanket of the switching variables are still its parents and hence for individual node X_{i,t_i} , $I(X_{i,t_i} | Pa(X_{i,t_i} | X_{i,t_{i-1}}))$ is true, where $Pa(X_{i,t_i})$ are parent set of X_{i,t_i} . However, for primary inputs, we have make sure that $P(X_{k,t_{i+1}}^p | X_{k,t_i}^p)$ handles the switching constraints properly. In sequential circuit, each input signal values are considered random, however, for modeling the the switching variables the temporal edges in inputs are essential and eliminates the need for explicit temporal dependencies of the intermediate random variables.

We call this graph structure as the temporal dependency model or TDM. Fig. 1(c) shows the TDM for the example sequential circuit in Fig. 1 (a); we just show two time slices here. The dash-dot edges shows the second type of edges mentioned above, which couples adjacent DAGs. We have X_2 as input and X_1 as the present state node. Random variable X_6 represents the next state signal. Note that this graph is a DAG.

The joint probability function is modeled by a Bayesian network as the product of the conditional probabilities defined between a node and its parents in the TDM structure: $P(x_v | Pa(X_v))$. These conditional probabilities can be easily specified using the circuit logic. There are three basic types of conditional probability specifications: (i) internal lines, (ii) primary input lines, and (iii) state lines. For the internal lines, the specification follows the gate logic. For state lines, the conditional probability models the logic of a buffer, as shown in Table 1(b). For primary input lines, the conditional probabilities models the switching constraints between two time instants, as listed in Table 1(a). For instance, if the primary line switched from 0 to 1, then at the next time slice the line can either switch from 1 to 0 or remain at 1. Since, we are considering random inputs, we distribute the probabilities equally between these two options. For correlated inputs, these conditional probabilities will have to be adjusted.

Table 1. Conditional probability specification between (a) primary input line switchings at consecutive time instants, and (b) state line switchings at consecutive time instants

(a)					(b)				
X_{k,t_i+1}^p				X_{k,t_i}^p	X_{k,t_i+1}^s				X_{k,t_i}^s
x_{00}	x_{01}	x_{10}	x_{11}		x_{00}	x_{01}	x_{10}	x_{11}	
0.5	0.5	0	0	x_{00}	1	0	0	x_{00}	
0	0	0.5	0.5	x_{01}	0	1	0	x_{01}	
0.5	0.5	0	0	x_{10}	0	0	1	x_{10}	
0	0	0.5	0.5	x_{11}	0	0	0	x_{11}	

5 Inference in TDM

Stochastic sampling algorithms are approximate BN inference schemes. Probabilities are inferred by a complete set of samples or instantiations that are generated for each node in the network according to the conditional probability table which stores the conditional probability of a random variable given its parents. In these sampling schemes each sample determines the posterior probability of the underlying model for the remaining samples. The probability of random variable is proven to converge to the correct values given enough time. The salient features of these algorithms are: (1) They scale extremely well for larger systems making them a target inference for nano-domain billion transistor scenario and (2) They are any-time algorithm, providing adequate accuracy-time trade-off and (3) The samples are not based on inputs and the approach is input pattern insensitive.

For large circuits, Probabilistic Logic Sampling (PLS) [6] which uses local message passing and stochastic sampling, is appropriate. This method scales well with circuit size and is proven to converge to correct estimates. These classes of algorithms are also anytime-algorithms since they can be stopped at any point of time to produce estimates. Of course, the accuracy of estimates increases with time.

Probabilistic Logic Sampling(PLS) developed by Henrion in 1988 is credited to be the first stochastic sampling method for inferencing Bayesian Networks [6]. In this method sampling is performed in the forward direction(from parents to children). The algorithm works as follows,

- In the circuit which is represented as a Bayesian network each node is selected in top-down fashion and they are sampled.
- While inferencing the Bayesian network the samples are grouped into sets and the observed value

in each sample in a set is compared with the corresponding evidence values.

- If they are inconsistent with each other the whole sample set is discarded.
- The same method is repeated with each sample set.
- In the selected sample set the belief distributions are calculated by averaging the frequencies with which the relevant events occur.

As compared to the computational merits, this method also has some disadvantages. Since it is based on forward sampling, the evidence that have already occurred cannot be accounted until the corresponding variables are sampled. The occurrence of unlikely evidence can result in rejection of large number of samples thereby hindering the performance of this method. Due to this PLS is always considered to be better without evidence.

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

6 Experimental Results

We have used the sequential circuits from the ISCAS89 benchmark suite to verify our method. To generate the simulation results the circuits were simulated for 1000000 test vectors. The sequential circuits are modeled as a DBN with 3 time-slices. A startup simulation with 50 random test vectors was performed and these startup estimates of the present state lines are given to the first time-slice of the DBN, i.e. these are the prior probabilities of the state lines. The priors for the primary input lines in the first time-slice of DBN was chosen to be unique, i.e. equally probable switching states. The approximate computation of the DBN was done by a tool named "GeNIe" [5]. The tests were performed on a Pentium IV, 2.00GHz, Windows XP computer.

Table 2. Dynamic Bayesian network modeling using PLS[6] for sequential circuits (3 time slices).

Circuits	μ_E	\max_E	Time (s)	% of nodes $> \mu + 2\sigma$
s27	0.028	0.092	0.016	5.88
s208	0.014	0.224	0.281	9.02
s382	0.000	0.082	0.750	7.14
s444	0.005	0.067	0.843	3.90
s526	0.002	0.048	1.234	1.84
s713	0.009	0.067	1.968	4.70
s820	0.002	0.042	2.125	4.17
s953	0.012	0.185	1.922	7.95
s1196	0.001	0.043	2.735	5.17
s1238	0.003	0.035	2.703	5.00
s1423	0.012	0.114	3.266	6.02
s5378	0.001	0.389	23.128	4.98
s15850	0.003	0.434	146.992	3.07

Table 2 lists the error statistics of the circuits by PLS inference scheme. A sample size of 1000 samples was considered for both the schemes. Switching error is the error between the in-house logic simulation and the switching estimates obtained from Bayesian inference. We tabulate both the average error, μ_E , and maximum error, \max_E , over all the nodes in column 2 and column 3 in both Table 2(a) & (b). We also list the percentage of nodes with switching error above 2 standard deviations from the mean error. The fourth column in both Table 2(a) & (b) indicate the run-time of the circuit with the specific inference scheme. The listed elapsed times are obtained by the *ftime* command in the WINDOWS environment, and is the sum of CPU, memory access and I/O time.

We see that the mean error is extremely small for most benchmark circuits and even for larger benchmarks like *s5378*, *s15850*. Even the maximum errors for most circuit are low. However, for some circuits, i.e. *s208*, *s953* and *s5378*, the maximum seem to be high, but these errors seem to be isolated to a few nodes as is seen from the low fraction of nodes with error above 2σ . In most cases, only 5% of the nodes exceed this error bound. In *s208*, we see that % of nodes in $\mu + 2\sigma$ range is around 9%. We also found the accuracy of our model is excellent even for larger benchmark like *s15850* ($\mu_E = 0.004$)

In Table 3, we show the modeling for three time slices versus ten time slices. We observe that ten time slices do not enhance the quality of estimates. This shows that third order temporal models are good enough for our benchmarks, which matches with the observations made in [9, 11].

Thus, we have successfully modeled the switching activity of sequential circuits by using a graphical dependency model which is compact and dependency pre-

Table 3. Switching activity estimation for two different time slices (3,10).

Circuits	3 Time slices			10 Time slices		
	μ_E	\max_E	Time(s)	μ_E	\max_E	Time(s)
s27	0.015	0.068	0.047	0.018	0.078	0.172
s208	0.014	0.234	0.719	0.006	0.197	7.532
s298	0.015	0.169	1.422	0.010	0.170	13.87
s382	0.002	0.096	2.094	0.000	0.079	21.28
s444	0.003	0.083	2.734	0.005	0.062	26.30
s526	0.003	0.044	3.922	0.001	0.081	42.28

serving. We have tested the model successfully using two stochastic estimation methods and presented the results. The scope of the work is limited to zero delay and the future effort is towards modeling real delay.

References

- [1] V. Mihajlovic and M. Petkovic, "Dynamic Bayesian Networks: A State of the Art," *Technical Report, TR-CTIT-01-35*, 2001.
- [2] U.Kjaerulff, "dHugin: A Computational System for Dynamic Time-Sliced Bayesian Networks," *International Journal of Forecasting*, vol. 11, pp. 89–111, 1995.
- [3] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference", Morgan Kaufmann Publishers, Inc., 1988.
- [4] S. Bhanja and N. Ranganathan, "Switching Activity Estimation of VLSI Circuits Using Bayesian Networks," *IEEE Transactions on VLSI Systems*, vol. 11, no. 4, pp. 558–567, Aug.
- [5] "Graphical Network Interface"
URL <http://www.sis.pitt.edu/~genie/genie2>
- [6] M. Henrion, "Propagation of uncertainty by probabilistic logic sampling in Bayes' networks," *Uncertainty in Artificial Intelligence*, 1988.
- [7] F. Najm, S. Goel and I. Hajj, "Power Estimation in Sequential Circuits," *32nd ACM/IEEE Design Automation Conference*, pp. 635–680, 1995.
- [8] A. Ghosh, S. Devadas, K. Keutzer and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," *29th ACM/IEEE Design Automation Conference*, pp. 253–259, 1992.
- [9] C. Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain and B. Lin, "Power Estimation Methods for Sequential Logic Circuits," *IEEE Transactions on VLSI Systems*, vol. 3, no. 3, pp. 404–416, 1995.
- [10] G. I. Stamoulis, "A Monte-Carlo Approach for the Accurate and Efficient Estimation of Average Transition Probabilities in Sequential Logic Circuits," *IEEE Custom Integrated Circuits Conference*, pp. 221–224, 1996.
- [11] L. P. Yuan, C. C. Teng and S. M. Kang, "Statistical Estimation of Average Power Dissipation in Sequential Circuits," *34th Design Automation Conference*, 1997.
- [12] Z. Chen and K. Roy, "An Efficient Statistical Method to Estimate Average Power in Sequential Circuits Considering Input Sensitivities," *ASIC Conference and Exhibit*, pp. 189–193, 1997.
- [13] J. N. Kozhaya and F. N. Najm, "Power Estimation for Large Sequential Circuits," *IEEE Transactions on VLSI Systems*, Vol. 9-2, pp. 400–406, 2001.
- [14] V. Saxena, F. N. Najm and I. N. Hajj, "Estimation of State Line Statistics in Sequential Circuits," *ACM Transactions on Design Automation of Electronic Systems*, vol. 7, no. 3, pp. 455–473, 2002.
- [15] D. Marculescu, R. Marculescu and M. Pedram, "Sequence Compaction for Probabilistic Analysis of Finite-State Machines," *34th Design Automation Conference*, pp. 12–15, 1997.