Any-time Probabilistic Switching Model using Bayesian Networks

Shiva Shankar Ramani
Dept. of Electrical Engineering
University of South Florida
Tampa, Florida 33620.
sramani@eng.usf.edu

Sanjukta Bhanja
Dept. of Electrical Engineering
University of South Florida
Tampa, Florida 33620.
bhanja@eng.usf.edu

ABSTRACT

Modeling and estimation of switching activities remain to be important problems in low-power design and fault analysis. A probabilistic Bayesian Network based switching model can explicitly model *all* spatio-temporal dependency relationships in a combinational circuit, resulting in zero-error estimates. However, the space-time requirements of exact estimation schemes, based on this model, increase with circuit complexity [1, 2]. This paper explores a non-simulative, Importance Sampling based, probabilistic estimation strategy that scales well with circuit complexity. It has the any-time aspect of simulation *and* the input pattern independence of probabilistic models.

1. INTRODUCTION

Due to drastic growth in semiconductor technology, the number of gates per chip have increased enormously. This increase in chip density, together with decrease in feature sizes have made power dissipation a major issue in VLSI circuits. Average power consumption in circuits depends on three factors: static leakage power, short circuit power, and dynamic power. As device sizes and transistor threshold scales down, power lost due to the leakage current will increase and will become sizable component of total power dissipation. However, the impact of dynamic power on total power consumption will remain to be substantial. The dynamic power consumed at the gate is given by $0.5V_{dd}^2 f_{clk} C_{load} Sw(x)$, where V_{dd} is supply voltage, f_{clk} is clock frequency, C_{load} is the load capacitance, and $Sw(x)$ is the switching activity at the node. Thus, modeling *and* estimation of switching activities remain to be important problems in low-power design and fault analysis.

This work is concentrated at the logic or gate level switching activity estimation. Estimation of switching activity requires the knowledge of input statistics, connectivity of the circuit, the correlation between nodes, the gate type, and the gate delays, ultimately making the estimation process a complex procedure. The correlations among the nodes affect switching activity. It has been established that for zero-delay model of a combinational circuit, only first order temporal correlation is exhibited [11], because signals possess first order Markov property. Thus, it is sufficient to consider just first order temporal correlation, but all high order spatial correlations to model *all* spatio-temporal dependencies in the combinational circuit. The non-simulative probabilistic techniques [4, 5, 7, 8] use knowledge about input statistics to probabilistically estimate the switching activity of internal nodes making the technique pattern-insensitive. In a later effort to capture higher order correlation approximately, Marculescu *et al.* in [6] handled higher order correlation as a composition of pair-wise correlations.

Recently, we proposed a novel model [1, 2], for switching activity estimation in combinational circuits using Probabilistic Bayesian Networks [3], that captures both the temporal and spatial dependencies in a comprehensive manner, resulting in zero-error estimates. Bayesian Networks are a Directed Acyclic Graph (DAG) representations, whose nodes represent random variables and the links denote direct dependencies, quantified by conditional probabilities of a node given the state of its parents. The DAG structure models the joint probability over a set of random variables in a compact manner. Bayesian Networks are exciting probabilistic inferencing models for VLSI circuits, particularly due to the following reasons.

1. Like any causal models, such as the traditional Binary Decision Diagrams (BDDs), Bayesian Networks models conditional independencies. However, Bayesian network has the least complex network structure in that is a minimal, compact independence map (I-map) of all the independencies among the underlying random variables. This minimal DAG structure is exploited to construct efficient probabilistic updating.
2. It can accommodate input correlation, temporal, and spatial correlations efficiently.
3. Bayesian Networks are unique probabilistic causal model in capturing the induced dependence between independent parents of a node given an observed state for the node.

Bayesian network models of switching activity are inherently zero-error models. However, the space-time complexity of exact estimation schemes increase with circuit complexity. For instance, the inference scheme that we used in [1, 2], which was a cluster based scheme, resulted exact estimates, however, it was memory intensive. So, for complex circuit, we had to resort to partitioning schemes, resulting in an approximate model of the switching activities in terms of a set of loosely coupled cascaded Bayesian Networks. This model produced estimates with low *mean* error, but due to coupling losses at the boundary nodes, it resulted in larger *standard deviation* and *maximum* error.

From a design point of view, it is sometimes desirable to have an estimation strategy where one can easily trade-off between time and accuracy, essentially an any-time estimation algorithm. This

ISLPED'04, August 9–11, 2004, Newport Beach, California, USA.

is not possible with the current inference scheme. For these reasons, in this paper, we explore three Stochastic Importance Sampling schemes: Probabilistic Logic Sampling (PLS) [12], Adaptive Importance Sampling (AIS) [9] and Evidence Pre-propagated Importance Sampling (EPIS) [10] for Bayesian updating. These algorithms combine the any-time feature of simulative approaches and input pattern independence of probabilistic approaches. Experimental results with ISCAS benchmark shows order of magnitude reduction in maximum error, standard deviation specially for larger benchmarks with significantly low time, especially for the PLS scheme. Moreover, the EPIS and AIS algorithms also facilitates probabilistic diagnosis or backtracking, which cannot be done by any current probabilistic models used in VLSI.

2. APPROACH

In this section, we discuss the fundamental modeling issues relevant to Bayesian Network. Interested reader is recommended to read [1, 3] for detailed understanding. As we mentioned before, Bayesian Networks are compact graphical probabilistic model for the underlying joint probability distribution function. Each node in the DAG structure is a random variable representing switching and can have four states ($0 \rightarrow 0,\ 0 \rightarrow 1,\ 1 \rightarrow 0,\ 1 \rightarrow 1$) for complete capture of temporal dependence under zero-delay scenario. Edges in the DAG denotes cause and effect relationship in the probabilistic model and is quantified by the conditional probability of a child node given its parents.

We define conditional independence in Eq. 1 with respect to the probabilistic model as follows:

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

$$P(x|y,z) = P(x|z) \text{ whenever } P(y,z) > 0 \tag{1}$$

For the graphical model, the conditional independence is studied by the concept of d-separation.

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

Definition 3: A DAG D is said to be an I-map of a dependency model M if every *d-separation condition* displayed in D corresponds to a valid conditional independence relationship in M, i.e., if for every three disjoint sets of vertices X, Y and Z, we have, $< X|Z|Y > \Rightarrow I(X,Z,Y)$.

Note that the Definition 3 holds the unifying feature of the graph based probability model in a way that connects the DAG D to the probabilistic model P. In Bayesian Networks, we not only suggest that DAG D is a dependency model for P (because all the d-separations in D imply a conditional independence in P), but also the notion of a compact minimal representation is built in. Let us consider the example of a probabilistic model P over four random variables $\{X_1, X_2, X_3, \textit{ and } X_4\}$ as shown in Figure 1. Note that, the DAG in Figure 1a, all the nodes are considered independent and hence I-map of D is greater than that of P which indicates that D under-represents P. In Figure 1d, the I-map of D is less than that of P as D is a complete DAG exhibiting maximum dependencies. This model would generate accurate results but are over-representation and hence the computation efforts would be large. A Bayesian Network has to be the DAG where the I-map for DAG matches the I-map of the P and hence it is the exact representation that is minimal in structure.

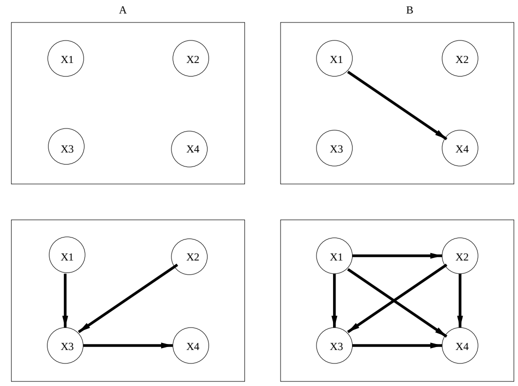

Figure 1: Bayesian Networks: Marriage between Graphical and Probabilistic Models.

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

Eq. 2 denotes the exact probabilistic model over random variables and using conditional independencies (in Eq. 3, we can arrive at the minimal factored representation shown in Eq. 4 which is the probabilistic model of Bayesian Network.

$$p(x_i|x_{i-1}, x_{i-2}, \cdots, x_1) = p(x_i|Pa(x_i)) \tag{3}$$

$$P(X) = \prod_{k=1}^{m} P(x_k|Pa(x_k)) \tag{4}$$

3. STOCHASTIC INFERENCE

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 [9] 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. The classes of algorithms selected here are known as Importance Sampling algorithms [9, 10, 12] which are not only good predictors or estimators, predicting the behaviors of descendent nodes (intermediate ones) given some properties of the primary inputs, but also accurate diagnostic tool that would relate possible pattern of the inputs given a particular set of behavior of any internal nodes.

Given information about evidences on any nodes and conditional probability table of each node, an Importance Sampling algorithm generates sample instantiations of all other nodes in the network. These instantiations can be used to form final estimates. Importance Sampling can be better understood by considering an approximate computation of an integral J, where

$$J = \int_{\Theta} b(X)dX \tag{5}$$

Let $b(X)$ be a function of k variables $X = (X_1, X_2, ...X_k)$ over a domain $\Theta \subset R^k$. An importance function $i(X)$ can be introduced in this integral, $J = \int_{\Theta} \frac{b(X)}{i(X)} i(X) dX$. The importance function $i(X)$ is a probability density function such that $i(X) > 0$ for any $X \subset \Theta$. After sampling the importance function over M instantiations $X_1, X_2, \ldots . X_M$, the approximate value of the integral is calculated as follows,

$$\hat{(J)} = \frac{1}{M} \sum_{i=1}^{M} \frac{b(X_i)}{i(X_i)} \tag{6}$$

The variance of $\hat{(J)}$ is proportional the variance of $i(X)$. In order to get an optimal importance function that can compute the integral, the variance of $i(X)$ should be zero provided $b(X) > 0$. Thus, the optimal importance function is represented as, $i(X) = \frac{b(X)}{J}$. The main goal of the Importance Sampling algorithm is achieving the importance function. The stochastic sampling strategy works because in a Bayesian Network the product of the conditional probability functions for all nodes is the optimal importance function. Because of this optimality, the demand on samples is low.

3.1 Probabilistic Logic Sampling

Probabilistic Logic Sampling (PLS) is the first and the simplest sampling algorithms proposed for Bayesian Networks [12]. The flow of the algorithm is as follows:

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

3.2 Adaptive Importance Sampling

In Adaptive importance sampling (AIS) scheme [9], the joint probability function that is modeled by a BN can be expressed as the product of the conditional probability of the nodes given its parent nodes is proven to be an optimum importance function. The steps for this algorithm are presented below:

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

3.3 Hybrid Scheme

For large circuits, a hybrid scheme, specifically the Evidence Pre-propagated Importance Sampling (EPIS) [10], which uses local message passing and stochastic sampling, is appropriate. This method scales well with circuit size and is proven to converge to correct estimates. These classes of algorithms are also anytime-algorithms since they can be stopped at any point of time to produce estimates. Of course, the accuracy of estimates increases with time.

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

$$P(X|E) = \prod_{k=1}^{m} P(x_k|Pa(x_k, E)) \tag{7}$$

This importance function can be approximated as

$$P(X|E) = \prod_{k=1}^{m} \alpha(Pa(X_k))P(x_k|Pa(X_k))\lambda(X_k) \tag{8}$$

where $\alpha(Pa(X_k)) = P(x_k|E^+)$ and $\lambda(X_k) = P(E^-|x_k)$, with E^+ and E^- being the evidence from above and below, respectively, as defined by the directed link structure. Calculation of λ is computationally expensive and for this, Loopy Belief Propagation (LBP) [13] over the Markov blanket of the node is used. Yuan *et al.* [10] proved that for a poly-tree, the local loopy belief propagation is optimal. The importance function can be further approximated by replacing small probabilities with a specific cutoff value.

The above set of stochastic sampling strategies discussed in subsection 3.1, 3.2, and 3.3 work because in a Bayesian Network the product of the conditional probability functions for all nodes is the optimal importance function. Because of this optimality, the demand on samples is low. We have found that just thousand samples are sufficient to arrive at good estimates for the ISCAS85 benchmark circuits. *Note that this sampling based probabilistic inference is non-simulative and is different from samplings that are used in circuit simulations.* In the latter, the input space is sampled, whereas in our case both the input and the line state spaces are sampled simultaneously, using a strong correlative model, as captured by the Bayesian Network. Due to this, convergence is faster and the inference strategy is input pattern insensitive.

4. RESULTS AND CONCLUSIONS

We experimented with the combinational circuits from the ISCAS85 benchmark suite. We first mapped the ISCAS circuits to their corresponding DAG structured Bayesian Networks. The experimental set-up of "GeNIe" [14], a graphical network interface is used for our experimentation. The tests were performed on a Pentium IV, 2.00GHz, Windows XP computer. For comparison, we performed zero-delay logic simulation on the ISCAS85 benchmark circuits, which provides accurate estimates of switching. The total elapsed time reported in all the experiments is the sum of CPU, memory access and I/O time.

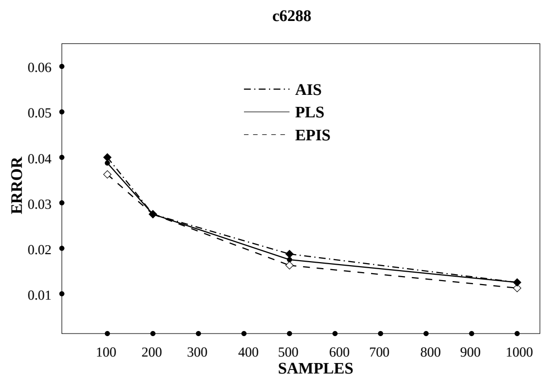

Figure 2: Graph showing the time accuracy trade off for c6288

Table 1: Experimental results comparing Approximate Cascaded Bayesian Network model and Probabilistic Logic Sampling

	Approx. CBN model [2]				PLS: 1000 samples			
	μ_E	σ_E	Mx_E	T(s)	μ_E	σ_E	Mx_E	T(s)
c432	0.00	0.02	0.28	3	0.00	0.00	0.04	0.40
c499	0.00	0.00	0.00	9.03	0.00	0.01	0.04	0.45
c880	0.00	0.00	0.04	2.52	0.00	0.01	0.05	1.05
c1355	0.00	0.00	0.09	1.81	0.00	0.01	0.06	1.75
c1908	0.00	0.01	0.15	10.70	0.00	0.01	0.05	2.7
c3540	0.00	0.04	0.26	18.86	0.00	0.00	0.04	5.96
c6288	0.01	0.04	0.37	38.75	0.00	0.01	0.06	11

Table 1 shows the mean, standard deviation, maximum error and the time elapsed for the ISCAS circuits with PLS and we compare the results with that obtained using the prior approximate Cascaded Bayesian Networks (CBN) [2]. Columns 2, 3, 4 and 5 in this table represents the mean error (μ_E), standard deviation of the error (σ_E), maximum error (Mx_E), and the elapsed time (T) for switching activity. It can be easily seen that even though good mean errors are obtained by approximate CBN methods, the stochastic PLS provides better estimates in terms of mean, standard deviation and shows significant improvement over the maximum error and computational time.

Tables 2 and 3 show the error statistics for predictive as well as diagnostic inference using Adaptive Importance Sampling and Evidence Pre-propagation Importance Sampling for 500, 1000 samples. Comparison of both the tables show the two algorithms converge close to accurate estimates within 500 samples. The mean and standard deviation of the error and the maximum error are extremely low for both the models even for larger benchmark circuits like c3540, c6288 even though EPIS is more efficient in terms of time. Note that both AIS and EPIS methods are efficient in probabilistic backtracking (from known observation to unknown cause) and outperforms PLS as well as the Approximate Cascaded Bayesian Network methods.

Figures 2, corresponding to, c6288 benchmark circuits, respectively, show the variation of errors, obtained using AIS, EPIS and PLS. Analysis of the graph shows that the estimates converge faster within a small sample space and estimates can always be formed even when the sample space is small or insufficient (any-time).

In this paper, we have demonstrated the results of the estimated switching activity using various any-time stochastic sampling inference algorithms namely EPIS, AIS, and PLS. We find that PLS yields the best accuracy-time tradeoff if used under predictive situation. In diagnostic situation, cases when evidence is unlikely, EPIS and AIS algorithms would yield accurate estimates. The present scope of this model is limited to zero-delay scenario, which we plan to address in future. We conclude that the Bayesian Network based modeling of switching activity and inference yields higher accuracy in significantly lower time.

Table 2: Experimental results using AIS algorithm for various samples.

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

Table 3: Experimental results using EPIS algorithm for various samples.

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

5. REFERENCES

[1] S. Bhanja and N. Ranganathan, "Switching Activity Estimation of VLSI Circuits using Bayesian Networks", *I*EEE Trans. on VLSI, vol. 11, no. 4, pp. 558-567, August 2003.

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

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

[4] F.N. Najm, "Transition Density: A New Measure of Activity in Digital Circuits", *IEEE Transaction on CAD*, vol. 12-2, pp. 310-323, February 1993.

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

[6] R. Marculescu, D. Marculescu, and M. Pedram, "Probabilistic Modeling of Dependencies During Switching Activity Analysis", *IEEE Trans. on CAD*, vol. 17-2, pp. 73-83, February 1998.

[7] B. Kapoor, "Improving the Accuracy of Circuit Activity Measurement", *Design Automation Conference*, pp. 734-739, 1994.

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

[10] C. Yuan, and M. J. Druzdzel, "An Importance Sampling Algorithm Based on Evidence Pre-propagation", *the 19th Annual Conf. on Uncertainty on AI*, pp. 624-631, 2003.

[11] P.Schneider, and U.Schlichtmann, "Decomposition of Boolean Functions for Low Power Based on a New Power Estimation Technique", *Int'l Workshop on Low Power Design*, pp. 123-128, 1994.

[12] M. Henrion, "Propagating Uncertainity in Bayesian Networks by Probabilistic Logic Sampling", In *Uncertainity in AI 2*, pp. 149-163, New York, N.Y., 1988.

[13] K. Murphy, Y. Weiss, and M. Jordan, "Loopy Belief Propagation for Approximate Inference: An Empirical study", *Conf. on Uncertainity in AI(UAI-99)*, pp. 467-475, 1999.

[14] "GeNie", URL http://www.sis.pitt.edu/~genie/genie2