# Dependency Preserving Probabilistic Modeling of Switching Activity using Bayesian Networks

Sanjukta Bhanja
Dept. of Computer Science and Engineering
Center for Microelectronics Research
University of South Florida
Tampa, Florida-33620.
bhanja@csee.usf.edu

N. Ranganathan
Dept. of Computer Science and Engineering
Center for Microelectronics Research
University of South Florida
Tampa, Florida-33620.
ranganat@csee.usf.edu

## ABSTRACT

We propose a new switching probability model for combinational circuits using a **Logic-Induced-Directed-Acyclic-Graph**(LIDAG) and prove that such a graph corresponds to a **Bayesian Network** guaranteed to map all the dependencies inherent in the circuit. This switching activity can be estimated by capturing complex dependencies (spatio-temporal and conditional) among signals efficiently by local message-passing based on the Bayesian networks. Switching activity estimation of ISCAS and MCNC circuits with random input streams yield high accuracy (average mean error=0.002) and low computational time (average time=3.93 seconds).

## 1. INTRODUCTION

Switching activity estimation strategies, important for power estimation can be divided into two broad categories: (i) estimation by simulation and (ii) estimation by probabilistic techniques. Estimation by simulation [4], [5], though time consuming, is extremely accurate. Probabilistic techniques [10], [13], [14] are fast and tractable but typically involves assumptions about joint correlations.

In this paper, we model the switching in a combinational circuit using a probabilistic Bayesian Network [1, 2], which allows us to capture both the temporal and spatial dependencies in a comprehensive manner. Bayesian networks are directed acyclic graph (DAG) representations whose nodes represent random variables and the links denote direct dependencies, which are quantified by conditional probabilities of a node given the states of its parents. This DAG structure essentially models the joint probability distribution over the set of random variables under consideration in a compact manner. The attractive feature of this graphical representation of the joint probability distribution is that not only does it make conditional dependency relationships among the nodes explicit it also serves as a computational mechanism for efficient probabilistic updating. Bayesian networks have traditionally been used in artificial intelligence and image analysis. Their use in power estimation is new.


*DAC 2001*, June 18-22, 2001, Las Vegas,Nevada, USA


We first construct the Logic Induced Directed Acyclic Graph (LIDAG) based on the logical structure of the circuit. Each signal in the circuit is a random variable in the LIDAG that can have four possible states indicating the transitions from $0 \rightarrow 0, 0 \rightarrow 1, 1 \rightarrow 0, 1 \rightarrow 1$. Directed edges are drawn from the random variables representing switching of the inputs to the random variable for switching at the output of each gate. We prove that the LIDAG thus obtained is a Bayesian Network which is the minimal representation that captures all the independency mapping. The salient advantages of switching activity estimation by modeling it as a Bayesian Network are as follows.

1. *It is able to produce globally consistent estimates of switching activity that takes into account spatial correlation by local message passing.*

2. *It can accommodate input correlation, temporal, and spatial correlation efficiently.*

3. *After a compilation process that converts the Bayesian network into a junction tree of cliques, further computation time is small.* Thus, repeated computation of switching activity of the circuit with different input statistics does not require much time.

4. *Finally, a Bayesian network, in addition to pair-wise correlations [8], also models conditional independence amongst subset of variables, which allows it to cover a large class of probabilistic dependencies.*

## 2. BACKGROUND WORK

Non-simulative power estimation techniques have been proposed in the past which have reasonable accuracy and extremely fast estimation procedures. They can be purely non-simulative [11], [3], [10] or they can be statistically simulative[6]. In statistical simulation, the estimated power is highly input sensitive. Although, Monte Carlo simulation technique can use input selection efficiently [6], these techniques have problems when the estimation process has to be performed under correlated input streams.

The non-simulative statistical techniques use knowledge about input statistics to estimate the switching activity of internal nodes. In some of the pioneering works around this idea, Najm *et al.* [17] estimated the mean and variance of current using probability waveforms. In [11], the concept of transition density is introduced and is propagated throughout the circuit by Boolean difference algorithm. However, these methods have problems in handling correlation between nodes and hence, the estimates are inaccurate when

the nodes are highly correlated. An accurate way of switching activity estimation is proposed in [10] which has a high space requirement. Tagged probability simulation is proposed in [13], which is based on the local OBDD propagation. The signal correlations are captured by using local OBDDs. However, spatio-temporal correlation between the signals is not discussed. Kapoor [15] has modeled structural dependencies and Schneider *et al.* [18] used one-lag Markov model to capture temporal dependence and Tsui *et al.* [16] modeled spatial correlation. Pair-wise correlation between circuit lines were first proposed by Ercolani *et al.* in [12]. Marculescu *et al.* in [7], studied temporal, spatial and spatio-temporal dependencies by capturing pair-wise correlations. In a later effort to capture higher order correlation approximately, Marculescu *et al.* in [8] handled higher order correlation as a composition of pair-wise correlations. Schneider *et al.* [19] proposed another an approximate technique to model higher order spatial correlations.

The theoretical contribution of our work is that the joint probability function of a set of random variables is exactly mapped capturing *higher order correlations between the signals accurately* using Bayesian Network model. Earlier efforts either treat the distribution as a composition of pair-wise correlated signals between all signals [8, 12] or use an approximate solution for capturing spatial correlation [19]. Moreover, the Bayesian Network models *conditional independence of a subset of signals* unlike in [8]. Results show that the technique has high accuracy with low execution times, for combinatorial benchmark circuits.

# 3. SWITCHING ACTIVITY MODELING

It is known that in a combinational circuit, switching at a node has correlations with its own past values and with its neighbors in the circuit. Temporal correlation is due to the fact that switching in a node is dependent on its last value. Spatial correlation among nodes arise of the underlying logical connections.

In this section, we first present mathematical notions that have been proposed to capture dependencies amongst random variables. Second, we list the conditions under which a Bayesian network can be constructed to capture the dependencies. Based on these foundations, we prove that all the dependencies amongst the switching variables in a combinational circuit can be captured using a DAG (directed acyclic graph) structured Bayesian network that is derived from the underlying circuit structure.

The following discussion is fashioned after [2]. We begin with the definition of *conditional* independence among three sets of random variables.

**Definition 1:** Let U=$\{\alpha, \beta, \cdots\}$ be a finite set of variables taking on discrete values. Let $P(.)$ be the joint probability function over the variables in $U$, and let $X$, $Y$ and $Z$ be any three subsets (maybe overlapping) of $U$. $X$ and $Y$ is said to be *conditionally independent* given $Z$ if

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

Following Pearl [2], we denote this conditional independency amongst $X$, $Y$, and $Z$ by $I(X,Z,Y)$; $X$ and $Y$ are said to be *conditionally independent* given Z. A dependency model, $M$, of a domain should capture all these triplet conditional independencies amongst the variables in that domain. A joint probability density function is one such dependency model. The notion of independence exhibits properties that can be axiomatized by the following theorem.

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

$$I(X,Z,Y) \Rightarrow I(Y,Z,X) \quad \text{(symmetry)} \quad (2)$$

$$I(X,Z,Y \cup W) \Rightarrow I(X,Z,Y)\&(X,Z,W) \quad \text{(decomposition)} \quad (3)$$

$$I(X,Z,Y \cup W) \Rightarrow I(X,Z \cup W,Y) \quad \text{(weak union)} \quad (4)$$

$$I(X,Z,Y)\&I(X,Z \cup Y,W) \Rightarrow I(X,Z,Y \cup W) \quad \text{(contraction)} \quad (5)$$

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

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

**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 set of vertices $X$, $Y$ and $Z$ we have, $< X|Z|Y > \Rightarrow I(X,Z,Y)$.

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

Note that every joint probability distribution function $P$ over a set of variables represents a dependency model $M$ since it captures all the conditional independencies.

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

There is an elegant method of inferring the minimal I-map of $P$ that is based on the notion of a Markov blanket and a boundary DAG, which are defined below.

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

**Definition 7:** Let $M$ be a dependency model defined on a set $U = \{X_1, \cdots, X_n\}$ of elements, and let $d$ be an ordering $\{X_{d1}, X_{d2}, \cdots\}$ of the elements of $U$. The *boundary strata* of $M$ relative to $d$ is an ordered set of subsets of $U$, $\{B_{d1}, B_{d2}, \cdots\}$ such that each $B_i$ is a Markov boundary (defined above) of $X_{di}$ with respect to the set $U_i(\subset U) = \{X_{d1}, X_{d2}, \cdots, X_{d(i-1)}\}$, i.e. $B_i$ is the minimal set satisfying $B_i \subset U$ and $I(X_{di}, B_i, U_i - B_i)$. *The DAG created by designating each $B_i$ as the parents of the corresponding vertex $X_i$ is called a* boundary DAG *of $M$ relative to $d$.*

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

**Theorem 2:** Let $M$ be any dependency model satisfying the axioms of independence listed in Eqs. 2- 5. If graph structure $D$ is a boundary DAG of $M$ relative to ordering $d$, then $D$ is a minimal I-map of $M$.

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

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

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

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

**Proof:** Let us order the random variables, $\{X_1, X_2, \cdots, X_{N+n}\}$ such that (i) variables representing the switching at the input lines appear first followed by those representing the output lines of the gates, and (ii) if a line $O_i$ is an input to a gate whose output line is $O_k$ then the variable corresponding to line $O_i$ appear before $O_k$.

With respect to this ordering, the Markov boundary of a node, $X_i$, is given as follows. If $X_i$ represents switching of an input signal line, then its Markov boundary is the null set. And, since the switching of an output line is just dependent on the inputs of the corresponding gate, the Markov boundary of a variable representing an output line consists of just those that represent the inputs to that gate. In the LIDAG structure the parents of each node are its Markov boundary elements hence the LIDAG is a boundary DAG. And, by Theorem 2 listed above the LIDAG is a minimal I-map and thus a Bayesian network (BN).

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

## 4. QUANTIFYING THE LIDAG-BN

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

Let us consider the circuit with fives gates shown in Figure 1. We are interested in the switching at each of the 9 numbered lines in the circuit. Each line can take four values corresponding to the four possible transitions: $\{x_{00}, x_{01}, x_{10}, x_{11}\}$. Note that this way of formulating the random variable effectively models temporal correlation. The probability of switching at a line would be given by $P(X_i = x_{01}) + P(X_i = x_{10})$[1]. The LIDAG structure for the circuit is shown in Figure 2. Dependence among the nodes that are not connected directly is implicit in the network structures. For example, nodes $X_1$ and $X_2$ are independent of each other, however, they are *conditionally* dependent given the value of say node $X_9$. Or the transition at line 5, $X_5$, is dependent on the transitions at lines 1 and 2, represented by the random variables, $X_1$ and $X_2$, respectively. Thus, the transitions of line 5 are *conditionally independent* of all transitions at other lines *given* the transition states of lines 1 and 2.

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

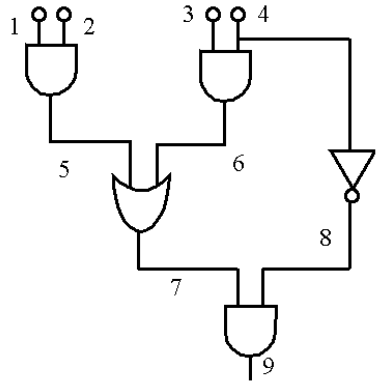


**Figure 1: A small combinational circuit.**

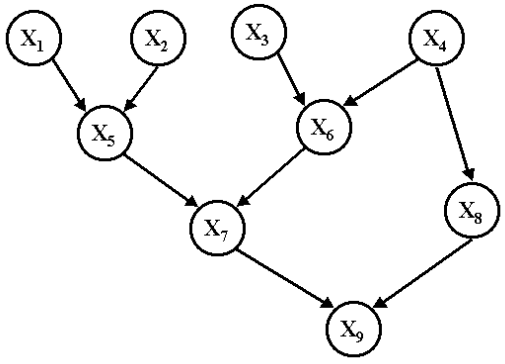


**Figure 2: Bayesian Network corresponding to the circuit in Figure 1.**

The joint probability function that is modeled by a Bayesian network can be expressed as the product of the conditional probabilities.

$$p(x_1, \cdots, x_N) = \prod_v p(x_v | x_{\text{parent}(v)}) \tag{6}$$

For the Bayesian network structure in Fig. 2 the corresponding joint probability density is given by the following factored form.

$$\begin{aligned} P(x_1, \cdots x_9) \quad = \quad & P(x_9|x_7, x_8) P(x_8|x_4) P(x_7|x_5, x_6) \\ & P(x_6|x_3, x_4) P(x_5|x_1, x_2) \\ & P(x_4) P(x_3) P(x_2) P(x_1) \end{aligned} \tag{7}$$

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

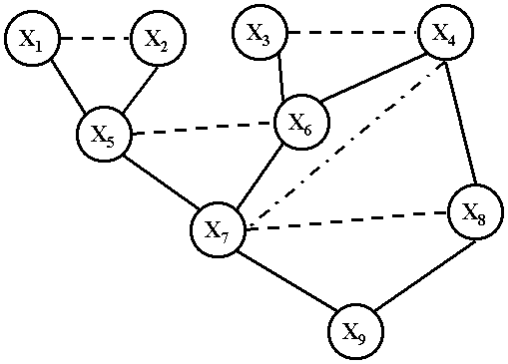


**Figure 3: Triangulated undirected graph structure that encodes that same dependencies as the DAG structure shown in Figure 2.**

The last four terms in the right hand side of Eq. 7 represent the statistics of the input lines. Given the statistics of the input lines, we would like to infer the probabilities of all the other nodes. A brute force way of achieving this would be to compute the marginal probabilities by summing over possible states, thus, $P(x_9, x_1) = \sum_{x_2,\cdots,x_8} P(x_1, \cdots, x_9)$. This, obviously, is computationally very expensive and, in addition, does not scale well. In the next section, we show how the structure of the Bayesian network can be used to efficiently compute the required probabilities.

## 5. BAYESIAN NETWORK COMPUTATIONS

It would be computationally convenient if we could compute probabilities in a Bayesian network by local message passing. Unfortunately, we cannot directly update a Bayesian network by local message passing if the underlying undirected graph structure has cycles. So, the first step of a propagation strategy is to transform the original DAG structure into a undirected tree structure whose nodes are subsets of the original random variables [1]. Such a tree is referred to as the *junction tree of cliques of random variables* and the process is referred to as the Bayesian network compilation process.

The first step of the compilation process is to create an undirected graph structure called the *moral graph* given the Bayesian network DAG structure. The moral graph represents the Markov structure of the underlying joint distribution [1]. In case of a DAG, which is the structure of a Bayesian network, a moral graph is obtained by adding undirected edges between the parents of a common child node and dropping the directions of the links. The second step of the compilation process triangulates the moral graph by adding additional links so that all cycles longer than 3 nodes are broken into cycles of three nodes. In our example,the dash line between $X_1$ and $X_2$ in Figure 3 is obtained during Moralization and the dash-dotted line between $X_4$ and $X_7$ in Figure 3 is added to the moral graph to triangulate it.

The next step in compilation is to detect cliques of nodes in the triangulated moral graph structuresuch that a junction tree can be formed between the cliques. Each clique is then clumped into a composite node representing the collection of nodes in the clique. The connection between the cliques is inherited from the triangulated moral graph structure. This tree of cliques is referred to as the junction tree of cliques. Figure 4 shows the junction tree for our running example. Every clique with an edge between them will have a non null set of nodes common between them, which is referred to as the separator set. These common variables play key role in evidence propagation. If there were two cliques that are not connected by an edge and still have common variables then these variables must be present in all the cliques in between the unique path between the two cliques to guarantee global consistency by local propagation. This is guaranteed by the triangulation step. As it can be seen that $C_3$ and $C_6$ both contain $X_7$, and $X_7$ is also present in all the cliques in the path from $C_3$ to $C_6$ namely in $C_1$ and $C_2$. This helps to preserve global consistency while updating by local message passing.

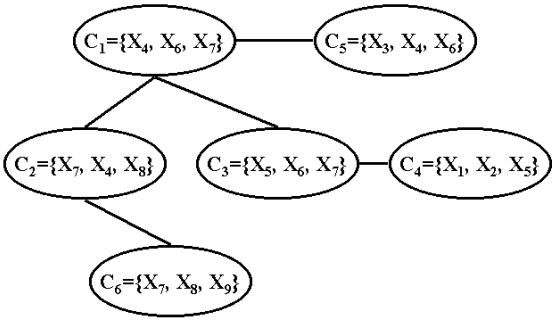


**Figure 4: Junction tree of cliques.**

It can be proven that the dependency properties of a DAG, which carry over to moral graphs, also are preserved in the triangulated moral graph [1]. The joint probability distribution that factorizes on the moral graph will also do so on the triangulated one since each clique in the moral graph is either a clique in this graph or subset of a clique.

After compilation, the probabilities are propagated through the junction tree just by local message-passing between the adjacent cliques.

Let us now consider two neighboring cliques to understand the key feature of the Bayesian updating scheme. Let two cliques *A* and *B* have probability potentials $\phi_A$ and $\phi_B$, respectively. Let *S* be the set of nodes that separates cliques *A* and *B*. The two neighboring cliques have to agree on probabilities on the node set *S* which is their separator. To achieve this we first compute the marginal probability of *S* from probability potential of clique *A* and then use that to scale the probability potential of *B*. The transmission of this scaling factor, which is needed in updating, is referred to as message passing. New evidence is absorbed into the network by passing such local messages. The pattern of the message is such that the process is multi-threadable and partially parallelizable. Because the junction tree has no cycles, messages along each branch can be treated independently of the others.

## 6. EXPERIMENTAL RESULTS

We mapped 14 ISCAS and 5 MCNC circuits to to their corresponding LIDAG structured Bayesian Networks. The conditional probabilities are pre-determined by the type of gate connecting the parents and the child. We have already discussed in Section 4 that each node in Bayesian network represents switching at a line in the circuit and can be in one of the four states $(x_{00}, x_{01}, x_{10}, x_{11})$. We used HUGIN's Bayesian Network tool for compiling the junction tree and propagating the probabilities. Circuits that are reasonably large, could not be modelled as a sigle BN. Multiple Bayesian Networks are used for modeling such circuits and probabilities are propagated between successive BNs. We also performed logic sim-

| | Error statistics over each node | | Overall % Error and time for all nodes | | |
|---|---|---|---|---|---|
| Circuits | $\mu_{Err}$ | $\sigma_{Err}$ | % Error | Elapsed Time (s) Total | Update |
| c17 | 0.0002 | 0.0004 | 0.02% | <.001 | <.001 |
| c432 | 0.0111 | 0.0300 | 2.08% | 1.33 | <.001 |
| c499 | 0.0002 | 0.0038 | 0.09% | 0.33 | <.001 |
| c880 | 0.0012 | 0.0088 | 0.57% | 1.24 | <.001 |
| c1355 | 0.0015 | 0.0189 | 0.41% | 2.29 | <.001 |
| c1908 | 0.0009 | 0.0090 | 0.22% | 6.09 | <.001 |
| c3540 | 0.0029 | 0.0400 | 0.79% | 8.62 | <.001 |
| c5315 | 0.0035 | 0.0269 | 0.80% | 15.08 | <.001 |
| c6288 | 0.0140 | 0.0465 | 3.34% | 18.70 | <.001 |
| c7552 | 0.0031 | 0.0460 | 0.70% | 18.26 | <.001 |
| alu4 | 0.0001 | 0.0198 | 0.86% | 0.54 | <.001 |
| malu4 | 0.0011 | 0.0204 | 0.23% | 0.21 | <.001 |
| max_flat | 0.0004 | 0.0005 | 0.10% | <0.001 | <.001 |
| voter | 0.0002 | 0.0005 | 0.04% | 0.11 | <.001 |
| b9 | 0.0004 | 0.0020 | 0.10% | 0.33 | <.001 |
| c8 | 0.0002 | 0.0017 | 0.05% | 0.93 | <.001 |
| count | 0.0001 | 0.0006 | 0.03% | 0.33 | <.001 |
| comp | 0.0000 | 0.0003 | 0.00% | 0.22 | <.001 |
| pcler8 | 0.0002 | 0.0007 | 0.03%6 | 0.11 | <.001 |

**Table 1: Experimental results on switching activity estimation by Bayesian network modeling for ISCAS'85 and MCNC'89 benchmark circuits for random input sequences.**

ulation providing "ground truth" estimates of switching.

We tabulate the results of switching activity estimation at circuit level by the LIDAG structured Bayesian networks in Table 1. We also performed simulation with pseudo-random inputs for comparison of the switching estimates. Columns 2 and 3 provide mean and standard deviation of the error of switching activity, which is the error between the switching activity estimated and switching activity obtained through simulation on each node, respectively. Moreover, we present the %error between the average switching activity estimated over all the nodes with average switching activity through simulation in column 4. The elapsed total time for Bayesian Network based estimation, which is time to compile and time to propagate, is shown in column 5 . We separately show the time for propagation of evidence in column 6. As it can be observed, the propagation time is extremely low irrespective of the size of the circuits.

The platform used here is a single processor DELL PC with 250 Meg RAM and 450 MHz clock speed. We want to emphasize that the time estimates are obtained from a WINDOW based timing feature (function ftime in VISUAL C++) which provides the total time spent in a function module which includes memory access, I/O time along with CPU time. In fact, while handling larger circuit, CPU time is not a good estimate of actual time due to high memory access needed by the algorithm.

As it can be observed from Table 1, the circuits which are (namely c17, comp, count, pcler8 etc.) modeled as a single Bayesian Network yield exact estimate and the standard deviation of error calculated on each node is extremely low. The time for estimation is also less. The circuits which are modeled by multiple Bayesian Networks, have standard deviation of the order of 0.02. The errors encountered in larger circuits are contributed by the loss of some correlations in the network boundaries. The result reported in this paper are based on preliminary segmentation scheme, while currently we are investigating an efficient segmentation technique that will reduce the standard deviation and the mean error. The maximum error that is encountered is around 2% for c432. The minimum error is zero. **Out of the 19 benchmark circuits 17 of them had less than 1% error and 2 of them have between 1% and 3.5% error**. It can also be observed that updating and propagating all the input evidences in almost all the circuits were performed in the order of 1 milliseconds. This is advantageous particularly if the switching activity has to be estimated for different input signal statistics. The circuits can be precompiled, only propagation has to be done for different input statistics.

In Table 2, we show some comparative statistics of time and accuracy estimates on the common benchmarks reported by [7] and by Schneider et. al. [19] for switching activity estimation. We also tabulate results obtained by [9] which is the revised version of the work [8] available in the web. We are unable to compare our work with that made by [8] since we do not have their results on pseudo-random inputs. At this point, we have to remember that these experiments are conducted on different machines with different speeds and architecture. For example, in [7], the estimation is performed in SUN SPARC 2 whereas in [19], estimation is performed under SUN SPARC 10. In all the previous works [7, 19, 8] the run time is reported in CPU seconds which is a drastic underestimate of total time with memory intensive implementation. However, we report run time is the total time in the module including memory access and I/O time. As it can be observed that BN modeling for most of the circuits performs better than the existing approximate modeling of switching activity. In some circuits, we obtained almost 10 times improvement in terms of accuracy compared to that of [7, 19, 8]. This suggests that there is need for modeling higher order correlations accurately. The standard deviation of error is not presented by Schneider et.al. [19]. The time to compute the switching activity is extremely fast for BN modeling. The elapsed time estimate is in most cases is in the order of hundred times faster compared to the work by Marculescu et. al. [7](in CPU sec) and on an average several times faster than the estimation by Marculescu [19, 9](in CPU sec). We are aware that these works are performed on different machines with varied CPU speed and different architecture. A real comparison is not possible but qualitative judgment can be drawn. We can see that Bayesian Network based estimation strategy is competitive both with respect to time and accuracy.

# 7. CONCLUSION

This paper introduces a switching activity estimation tool that encapsulates all the dependencies, both in the internal nodes and in the inputs, in reasonable time and with high accuracies. We have shown results of the estimated switching activity for pseudo-random inputs. The model handles spatio-temporal dependencies between the nodes. It also models conditional dependencies of the nodes. The proposed model is accurate and captures higher order correlation among lines. Our future effort focuses on the segmentation techniques, and input modeling for capturing spatial correlation at the primary inputs using the same BN model.

# 8. REFERENCES


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

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

[3] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits", *Proc. of the 29th DAC*, pp.253-259, June 1992.

[4] 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.

| Circuit | Marculescu et. al., 94 [7] | | | Schneider et. al., 96 *[19] | | | Marculescu et. al., 98 *[9] | | | Bayesian Networks | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mu_{Err}$ | $\sigma_{Err}$ | Total Time(s) | $\mu_{Err}$ | $\sigma_{Err}$ | Time CPU (s) | $\mu_{Err}$ | $\sigma_{Err}$ | Time CPU(s) | $\mu_{Err}$ | $\sigma_{Err}$ | Time CPU(s) |
| c17 | 0.012 | 0.02 | 0.35 | - | - | - | 0.012 | 0.02 | 0.04 | 0 | 0 | <0.001 |
| c432 | 0.013 | 0.02 | 276.43 | 0.016 | - | 11 | 0.028 | 0.04 | 11.82 | 0.011 | 0.03 | 1.33 |
| c499 | 0.005 | 0.01 | 519.56 | - | - | - | 0.013 | 0.01 | 10.57 | 0 | 0 | 0.33 |
| c880 | 0.016 | 0.03 | 320.11 | 0.006 | - | 34 | 0.013 | 0.02 | 18.35 | 0.001 | 0.01 | 1.24 |
| c1355 | 0.003 | 0 | 330.11 | 0.005 | - | 11 | 0.004 | 0 | 4.24 | 0.001 | 0.02 | 2.29 |
| c1908 | 0.004 | 0.01 | 489.23 | 0.01 | - | 8 | 0.009 | 0.02 | 12.69 | 0.001 | 0.01 | 6.09 |
| c3540 | 0.028 | 0.03 | 3307.11 | 0.014 | - | 65 | 0.03 | 0.04 | 60.86 | 0.003 | 0.04 | 8.62 |
| c5315 | - | - | - | - | - | - | - | - | - | 0.003 | 0.03 | 15.08 |
| c6288 | - | - | 4530 | 0.023 | - | 126 | 0.014 | 0.02 | 29.1 | 0.014 | 0.05 | 18.7 |
| c7552 | - | - | - | - | - | - | - | - | - | 0.003 | 0.05 | 18.26 |

**Table 2: Comparison of CPU time and Percentage error of different estimation techniques [7, 9, 19].**


[5] A. C. Deng, Y. C. Shiau, and K. H. Loh, "Time Domain Current Waveform Simulation of CMOS Circuits", *IEEE Intl. Conf. on CAD*, Santa Clara, CA, pp.208-211, Nov. 7-10,1988.

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

[7] 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.

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

[9] 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

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

[11] 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.

[12] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Testability Measures in Pseudorandom Testing", *IEEE Transactions on CAD*, vol.11, pp. 293-318, June 1992.

[13] C.-S. Ding, C.-Y. Tsui, and M. Pedram, "Gate-Level Power Estimation Using Tagged Probabilistic Simulation", *IEEE Transaction on CAD*, vol.17-11, pp.1099-1107, November, 1998.

[14] K. Parker, and E. J. McCluskey, "Probabilistic Treatment of General Combinational Networks", *IEEE Trans. on Computers*, vol. C-24, pp.668-670, June 1975.

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

[16] C.-Y. Tsui, M. Pedram, and A. M. Despain, "Efficient Estimation of Dynamic Power Dissipation with an Application", *Proc. IEEE/ACM Intl. Conference on Computer Aided Design*, pp. 224-228, November 1993.

[17] F. N. Najm, R. Burch, P. Yang, and I. N. Hajj, " Probabilistic Simulation for Reliability Analysis of CMOS Circuits", *IEEE Transaction on CAD*, vol 9-4, pp.439-450, April 1990.

[18] 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.

[19] P. H. Schneider, U. Schlichtmann, and B. Wurth, "Fast power estimation of large circuits", *IEEE Design & Test of Computers*, vol. 13-1, pp. 70-78, Spring 1996.