Cascaded Bayesian Inferencing for Switching Activity Estimation with Correlated Inputs

Sanjukta Bhanja and N. Ranganathan, Fellow of IEEE

***Abstract*—**

In this work, we investigate the estimation of switching activity in VLSI circuits using a graphical probabilistic model based on Cascaded Bayesian Networks (CBN's). First, we develop a theoretical analysis for Bayesian inferencing of switching activity and then derive upper bounds for certain circuit parameters which, in turn, are useful in establishing the cascade structure of the CBN model. We formulate an elegant framework for maintaining probabilistic consistency in the interfacing boundaries across the CBN's during the inference process using a tree-dependent (TD) probability distribution function. A TD distribution is an approximation of the true joint probability function over the switching variables, with the constraint that the underlying Bayesian network representation is a tree. The tree approximation of the true joint probability function can be arrived at by using a Maximum Weight Spanning Tree (MWST) built using pairwise mutual information about the switching occurring at pairs of signal lines on the boundary. Further, we show that the proposed TD distribution function can be used to model correlations among the primary inputs which is critical for accuracy in modeling of switching activity. Experimental results for ISCAS circuits are presented to illustrate the efficacy of the proposed CBN models.

I. Introduction

Switching activity is one of the important parameters for power estimation and reliability analysis. The switching activity of a node is affected by various factors such as the connectivity of the circuit, the input statistics, the correlation among nodes (or lines), the gate type, and the gate delays, thus making the estimation process a complex procedure. It is well-known that switching activity depends on temporal, spatial, and spatio-temporal correlations exhibited by the signals, which could be internal nodes or primary inputs or state lines.

In this work, we propose a new switching probability model for combinational circuits based on the concept of **Cascaded-Bayesian Networks** (CBN), capturing complex conditional dependencies over a set of random variables. Bayesian Network (BN) is a graphical representation of the joint probability function over a set of random variables in which nodes denote random variables and directed edges represent direct dependencies, quantified by the conditional probability of the child node given its parents. It is known that switching in a combinational circuit can be easily modeled as a Bayesian Network (BN) representation. Since Bayesian Network is an exact graphical representation of the underlying switching probability function, it captures higher order interdependencies among the switching variables completely rather than propagating the effects of low order correlations (usually pair-wise) in most probabilistic modeling.

In the BN structure, each random variable represents switching activity of a single line in the combinational circuit and can take four values corresponding to the four possible transitions: $\{x_{00}, x_{01}, x_{10}, x_{11}\}$ which ensures a lag-one Markov model to capture temporal effect completely in a zero-delay scenario [18]. The probability of switching at a line X_i would be given by $P(X_i = x_{01}) + P(X_i = x_{10})$. The conditional probabilities that are needed to quantify the dependencies in BN are obtained directly from the logic structure. Bayesian Inferencing is NP-hard and hence a single BN is not sufficient to model large circuits. This forces us, given the available computing constraints, to represent combinational circuits using the concept of Cascaded BNs.

The contributions of this work are many fold. First, we investigate ways to infer the computational complexity of Bayesian inferencing of switching activity, dependent on the maximum clique-size, through the construction of the*elimination set*, which is an ordered superset of the clique-sets known to have an unique property of running intersection a requirement for Bayesian inferencing by local message passing. Using this *elimination set*, which can be constructed in $O(n+e)$, (n and e are number of nodes and edges) we establish an upper bound on the complexity of the BN inference in terms of circuit parameters. The estimate of the upper bound, which we express in terms of maximum fan-ins and maximum fan-outs, can be used to partition the circuit into loosely coupled, cascaded BN representations. Second, we address the problem of maintaining consistency across the Cascaded BN representations during the inference process. Third, we address the problem of modeling the switching activity among *correlated* primary inputs. The problems of input correlation modeling and maintaining correlations across the CBN is essentially the same. For input modeling, we need a mechanism to represent the correlations among the inputs and for cascaded Bayesian modeling, we need to capture the correlations among the nodes of the boundaries between cascaded BN's. It is possible to elegantly address both these problems, similar in nature, with the same concept of tree-dependent distribution.

A tree-dependent (TD) distribution is an approximation of the true joint probability function over the switching variables, with the constraint that the underlying Bayesian network representation is a tree. The tree structure controls the computational complexity. The tree approximation of the true joint probability function can be arrived at using a Maximum Weight Spanning Tree (MWST) based on the pair wise mutual information between switchings at two signal lines [3]. Since the tree-dependent (TD) distribution can also be represented as a Bayesian Network, we can fuse this approximate tree representation over the primary nodes with the accurate BN-based representation over the internal nodes of the combinational circuit to form the TD-BN structure. This TD-BN structure can also be used to couple the BNs (as a CBN), in which the correlations among the boundary nodes are represented. Construction of CBN requires $O(n^2)$ computational effort. Also, if the compiled Bayesian Networks occupy most of the memory re-

This work is being supported in part by a grant from the National Science Foundation Grant No.CCR-0098103.

S. Bhanja and N. Ranganathan are with University of South Florida, Tampa, FL 33620, USA. E-mail: bhanja@eng.usf.edu, ranganat@csee.usf.edu

sources, computing mutual information itself takes significant computational time. Hence, we resort to a heuristic tree that will compute mutual information between boundary nodes only if they have common inputs or common children. We show how our proposed Approximate TD modeling lowers the mean error, standard deviation of error, and maximum error over a naive Bayesian network coupling, indicating that CBN is indeed a superior model, estimating switching activity accurately and uniformly over all the nodes.

II. Related Work

Switching activity estimation through probabilistic techniques [10], [13], [14], [5] are fast and tractable, but typically involve assumptions about joint correlations. Probabilistic 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 switching activity through probabilistic simulation and in a later work, Najm *et al.* [11] introduced the concept of transition density which is a measure of switching activity. However, these methods have been reported to yield less accurate estimates when the nodes are highly correlated. An improved switching activity estimation strategy based on OBDD was proposed by Bryant [10], however, it had high space requirements. Ghosh *et al.* [5] modeled temporal correlation effects in a real delay model. The computational complexity was, however, extremely high. Tagged probability simulation with a partial real delay model was proposed by Ding *et al.* [13], which was based on local OBDD propagation capturing first order spatial correlations.

Dependency modeling of switching activity has been performed by many, but only partially. Present formalisms are not able to account for all types of spatial dependencies. Kapoor [15] has modeled structural dependencies and Schneider *et al.* [18] used one-lag Markov model to capture temporal dependence. Tsui *et al.* [16] modeled first order spatial correlation efficiently. Schneider *et al.* [19] proposed a fast estimation technique based on ROBDD where an approximate solution is provided for spatial correlation to reduce time complexity of the exact approach. Pair-wise correlation between circuit lines were first proposed by Ercolani *et al.* [12]. Marculescu et al. [21], studied temporal, spatial dependencies jointly by a local OBDD that handles pairwise correlations. In a later pioneering work, Marculescu *et al.* [7], formulated higher order correlations by approximating them as a set of pair-wise correlations. Moreover, Marculescu *et al.* [7] was the first to study the effect of highly correlated input streams where they tried to capture the input correlation approximately. They also proved that correlations exhibited in the inputs have significant effect on switching activity throughout the circuit, and hence showed that power estimation assuming random inputs is not the correct picture.

Our focus in this paper is to analyze the theoretical aspects of Bayesian inferencing and establish upper bounds using circuit parameters, which help in determining the size and structure of each BN in the cascade. We formulate methods to capture the correlations among the primary inputs as well as the among the boundary nodes of CBN's using the concept of tree-dependent (TD) probability function. The proposed cascaded Bayesian inferencing model incorporates the above methods to improve the estimation accuracy without greatly increasing the computational cost.

III. Analysis of BN Inferencing

In this section, we provide a theoretical analysis of Bayesian inferencing for switching activity modeling. The purpose of this analysis is to decide whether a given circuit can be mapped to a single BN or not, given system constraints such as memory and CPU speed. Our goal is to partition the circuits into the least number of BNs keeping each of them as large as possible. This is because there is no loss of accuracy during the probability propagation within a BN while in cascaded BN's, some accuracy is lost at the boundaries in the coupling between the BN's even with the effort to reduce the loss by TD. This is a common challenge in probabilistic approaches when large circuits need to be partitioned for the computations. In the next section, we show how to minimize these coupling losses by reducing the number of instances when such losses can occur.

For the inference process, the Bayesian network structure is transformed first into a **moral graph (M)** by removing the directions on the links and mutually connecting the parents of a node. It should be noted that from this point onwards the directional properties of BN will not be needed in the inference process. This moral graph is then triangulated and a *junction tree* of cliques is formed, which is used for the cluster based inferencing. It is known from [22] that Bayesian network inference process based on the junction tree is limited by the product of the number of nodes and the exponential of the maximum clique size in the junction tree.

Since the triangularization of the moral graph is NP-hard, a minimum fill-in heuristic is used for triangularization from which *clique set* is obtained by constructing the intermediate *elimination set*. Hence, the maximum number of nodes in the cliques can be obtained by the maximum number of nodes in the *elimination set*. The triangularization is performed as follows: All the vertices of the moral graph are first unlabeled. An unlabeled vertex that has the minimum number of unconnected neighbors (only unlabeled neighbors) is chosen first. This vertex is then labeled with the highest available node number, say *i*, starting from a number equal to the total number of nodes. A set C_i, is then formed consisting of the selected vertex and its still unnumbered neighbors. Edges are filled in between any two unlinked nodes in this set C_i. Then the maximum available node number *i* is decremented by 1. This process is repeated until there is no unlabeled-numbered nodes. The resultant graph is guaranteed to be triangularized. Note that each C_i is a complete subgraph by construction and the set of these constitutes the cliques of the graph G. The generated sequence of cliques E=$\{C_i\}$'s is termed the *elimination set* of cliques of the graph.

In the moral graph shown in the left graph in Figure 1 (the dotted lines are achieved by connecting the parents during moralization), node X_9 is first selected since no fill-in edge is needed because all the neighbors (remember the moral graph is undirected) are already linked. This node X_9 is assigned the number 9 - the total number of nodes in the graph. The set C_9 is then formed by nodes $\{X_9, X_8, X_7\}$. The nodes X_8 and X_7 are not yet numbered. For the second cycle, the nodes X_8, X_7, X_6,

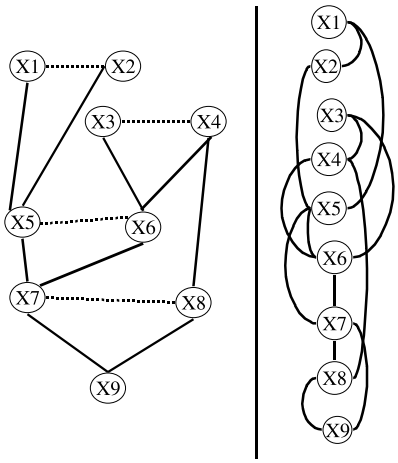


Fig. 1. Moral Graph M and its induced graph w.r.t. ordering d.

and X_4 cannot be selected as they each would require one fill-in edge amongst its neighbors, whereas the neighbors of X_3 do not require any fill-in edges. Hence X_3 is numbered 8 in our example and C_8 is formed by $\{X_3, X_4, X_6\}$. For the third cycle, we then select X_2, numbering it as 7 and forming $C_7 = \{X_2, X_1, X_5\}$. In the fourth cycle, node X_1 is assigned 6 and $C_6 = \{X_1, X_5\}$ is formed. We then select X_5, assign a number 5, and form $C_5 = \{X_5, X_6, X_7\}$. Node X_8 is assigned number 4, and $C_4 = \{X_8, X_7, X_4\}$ is formed. In this step, a fill-in edge between X_4 and X_7 is added. We then assign the number 3 to X_7, the number 2 to X_6, and the number 1 to X_4. The resultant *elimination set* $\{C_i\}$ obtained from our example is

$$\begin{aligned}\{C_1,\cdots,C_9\} \quad = \quad & \{\{X_4\},\{X_6\},\{X_7,X_6,X_4\},\\ & \{X_8,X_7,X_4\},\{X_5,X_6,X_7\},\{X_1,X_5\},\\ & \{X_2,X_1,X_5\},\{X_3,X_4,X_6\},\\ & \{X_9,X_8,X_7\}\}\end{aligned}$$

Definition : Let $G(V,E)$ contain p cliques. An ordering $[C_1, C_2, \cdots, C_p]$ possesses **running intersection** property if for every $j > 1, \exists i, i < j$ such that $C_j \cap (C_1 \cup C_2 \cdots \cup C_{j-1}) \subseteq C_i$.

By the construction of the *elimination set*, it possesses the *running intersection property*, namely, for C_4, $\{X_8, X_7, X_4\} \cap (\{X_4\} \cup \{X_6\} \cup \{X_7, X_6, X_4\}) = \{X_4, X_7\} \subseteq C_3$. This property is essential for BN inferencing and is solely responsible for fast inferencing through message passing between neighboring cliques.

It is shown in [1] that if $C_1, \cdots C_k$ is a sequence of sets having running intersection property and $C_t \subseteq C_p$ for some $t \neq p$ then the ordered set $C^{`} = \{C_1, \cdots C_{t-1}, C_p, C_{t+1}, \cdots, C_{p-1}, C_{p+1}, C_k\}$ also has running intersection property. By this property C_t can be eliminated for all $C_t \subseteq C_p, p \neq t$. Hence, the *elimination set* can be reduced to obtain the minimal ordered set of cliques called *Clique set* representing the triangularized graph completely. A junction tree between these cliques is then obtained and the probability for each clique is computed, requiring storage of the order of $O(n.4^{|C_{max}|})$ where n is the number of vertices and $|C_{max}|$ is the maximum clique size. In fact, updating probabilities of the single variables in the cliques can be performed in $O(p.4^{|C_{max}|})$ time where p is the number of cliques. Since, the number of cliques is generally much less than the number of vertices, memory requirement poses larger problem for BN inferencing. It can be easily concluded that since the *elimination set* possesses running intersection property, the maximum cardinality of the *elimination set* is the maximum cardinality of the *clique set*. Hence, even before we compute the probability function for the cliques and probabilistic updating, we can infer accurately the complexity of the inference by constructing the *elimination set*. It is obvious that the elimination set can be obtained by the minimum fill-in heuristics mentioned above in $O(n+e)$ where n is the number of vertices and e is the number of edges and hence the complexity of obtaining the elimination set is essentially $O(n^2)$.

Since the circuits are reasonably sparse, the elimination time is not too high for real benchmarks. In the in experimental section, we tabulate total compilation time in which the moralization, triangularization (elimination set construction is a part of this step) and clique joint probabilities are computed. The overall compilation time is less than 19 seconds and majority of this time is consumed in calculation of joint probabilities of the cliques.

For a large circuit, we compute $n.4^{|C_{max}|})$ and if it is larger than a threshold, we reduce the circuit. Obviously, we need a simpler bound for such partitioning and hence we develop $O(n)$ bounds on the maximum clique size and we discuss them in the next subsection.

A. Bounds for Circuit Structure

In this subsection, we use the concept of induced width [22] and obtain simple bounds based on induced width, which are computationally simple and easily obtained from circuit structure. These bounds serve as an approximate index for complexity and can be used to partition the circuits. We are concerned only with undirected graph as our analysis concerns moral graph, which is undirected.

Definition 1: The width w_x of a node x in an undirected graph G is defined as the number of neighbors of the node that precedes this node in a given order d of the graph nodes.

Definition 2: The width of an ordering $w_{G,d}$ is defined as the maximum width w_x in the undirected graph G, with respect to a particular order d.

Definition 3: The induced graph IG of an undirected graph G with respect to a particular ordering d can be obtained by processing nodes in the reverse order of d (last to first) and adding edges between each of the parents preceding to a node in the order d with each node.

For example, the induced graph of the example graph M shown in the left side of Figure 1 with respect to ordering $(X_9, X_8, X_7, X_6, X_5, X_4, X_3, X_2, X_1)$ is illustrated in the right hand side of figure. The width of any node with respect to this ordering can be obtained using the number of links directed downwards from those nodes.

Definition 4: The induced width $W_{G,d}$ of an undirected graph G with respect to a particular order d is the width of ordering of the induced graph IG. Hence,

$$W_{G,d} = w_{IG,d} \tag{1}$$

Definition 5: The induced width W_G of a graph is the *minimum* induced width of the graph (G) over *all* possible orderings.

$$W_G = \min W_{G,d} \forall d \tag{2}$$

Hence, for our analysis, for the undirected *Moral graph M*, we establish an upper bound for the induced width of the Moral graph W_M which is the minimum of $w_{IM,d}$ for all possible ordering d, where IM denotes the induced graph of the Moral graph M.

$$W_M = \min W_{M,d} \text{ and } W_{M,d} = w_{IM,d} \tag{3}$$

It is known [22] that the task of finding the *induced width of a graph, which entails considering all possible node orderings, is NP-hard*. Hence, estimating the exact induced width is not feasible. However, a rough estimate of it is needed for partitioning the circuits into BNs such that each BN is large enough, to be handled by the available resources and minimize error in the estimation process. Hence, we establish upper bounds on this induced width W_M that can be computed efficiently in terms of the circuit parameters.

Theorem 1: Given a combinational circuit, let I_i be the fan-in of the input of the gate with output i, O_i be the fan-out of the gate with output i, I_{max} be the maximum fan-in of the gates of the circuit, O_{max} be the maximum fan-out of the gates of the circuit, and $child(i)$ be the set of output signals of a gate/buffer where i is an input. Then, an upper bound U_{W_M} of the induced width of the *moral graph* of the BN is given by:

$$U_{W_M} = I_{max} + I_{max} O_{max} \tag{4}$$

A tighter upper bound Ut_{W_M} is given by:

$$Ut_{W_M} = \max_i (I_i + O_i + \sum_{j \in child(i)} I_j - 1) \tag{5}$$

Proof: From Eq. 3 one can easily observe that

$$W_M \leq W_{M,d}, W_{M,d} = w_{IM,d}, \text{and } w_{IM,d} \leq \max_{v \in M}[w_v] \tag{6}$$

The width of node i in moral graph M can be obtained with the information of node i in the original BN. Considering the worst case where all the links connected to the node i is preceding it in a particular ordering. We then have

$$w_i = I_i + O_i + \sum_{j \in child(i)} I_j - 1 \tag{7}$$

where I_i denotes the number of links to the node i's parents (edges in the BN), O_i denotes the number of links to the node i's children (edges in the BN), and $\sum_{j \in child(i)} I_j - 1$ denotes the number of extra edges added during moralization, since each node j will have a maximum of$I_j - 1$ number of parents preceding i, in the worst case ordering, which will be linked with i. Hence,

$$\max_i w_i = \max_i [I_i + O_i + \sum_{j \in child(i)} I_j - 1] \tag{8}$$

$$W_M \leq \max_i [I_i + O_i + \sum_{j \in child(i)} I_j - 1] \tag{9}$$

and

$$Ut_{W_M} = \max_i (I_i + O_i + \sum_{j \in child(i)} I_j - 1) \tag{10}$$

TABLE I
BN INFERENCE COMPLEXITY AND ITS UPPER BOUNDS.

BN from Circuits	Maximum Clique size	Ut_{W_M}	U_{W_M}
c432	7	10	20
c499	4	4	18
c880	5	8	18
c1355	5	14	26
c1908	10	16	34
c3540	5	6	14
c6288	5	5	8
c7552	3	8	14

Now, it is evident that $Ut_{W_M} \leq I_{max} + O_{max} + O_{max} I_{max}$ and hence simplifying $Ut_{W_M} \leq U_{W_M} = I_{max} + I_{max} O_{max}$. □

In Table I, we show the difference between the upper bound Ut_{W_M}, U_{W_M} and maximum clique size $(W_M + 1)$ for the BNs that are generated by randomly partitioning the ISCAS'85 circuits. In our BN modeling, to control complexity of representation, we replaced each node with fan-in greater than 2 by a combination of nodes with fan-ins of 2. Thus, the maximum fan-in for our BNs are 2. The fan-outs can, of course, be larger than 2.

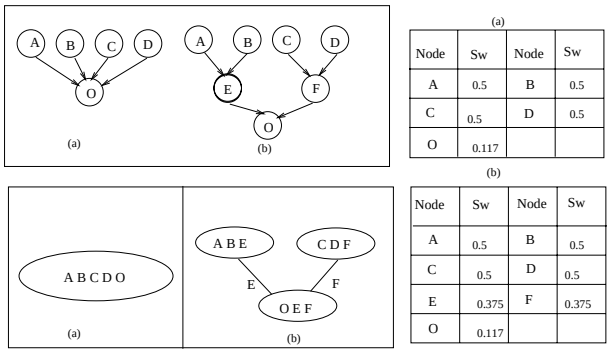
The upper bound estimate can be used to decide on the size of each BNs as we partition large circuits into cascaded BN. The memory required to compute using a BN will be $O(n.4^{W_M+1})$. We can easily compute both the upper bounds and number of nodes n to determine the size of the individual BN. In our approach we find that for most circuits the number of segments are on an average around 5 and for c6288 and c7552, we need a maximum of 18 individual BNs.

Note that the upper bounds are not tight. This is because we assume that all the fan-ins of the fan-outs of the node in consideration are ordered before the selected node. Hence by choosing BNs based on this bound, one might select smaller Bayesian Networks than the memory would allow. Performance would thus vary with different partitioning schemes. In fact, if the estimation is performed off-line, the designer might want to use the exact inference complexity calculation (O(n^2)) through the dimension of elimination set discussed earlier in this section. However, since our goal is to prove that CBN works better than BN for any generic partition (good or bad), we use the computationally inexpensive bounds.

B. Partitioning Strategies

In this paper, we did not address partitioning strategies comprehensively, in fact the purpose of introducing TD is to alleviate the effect of a bad partition and this paper concentrates on that effort. However, we provide the guidelines for best partitioning scheme that can generate best segmentation. We will also discuss partitioning method used here for experimentation.

The ideal partitioning requires that each Bayesian Network is large given a resource of the computing device and also where the boundary nodes between the Bayesian Networks are least correlated. The largeness of a Bayesian Network is measured in terms of spatial requirements for inference which is



(a)

Node	Sw	Node	Sw
A	0.5	B	0.5
C	0.5	D	0.5
O	0.117		

(b)

Node	Sw	Node	Sw
A	0.5	B	0.5
C	0.5	D	0.5
E	0.375	F	0.375
O	0.117		

Fig. 2. 4 input AND gate built through 2-input AND gates: probabilistic inference

$\tau = n.4^{|C_{max}|})$ where C_{max} is the maximum clique size and n is the number of nodes in the Bayesian Networks. Hence the partitioning problem through min-cut becomes *How can we partition a Bayesian Network into smaller loosely coupled ones such that each Bayesian Network has largest* τ *value (determined by the memory resource of the computing device) and the boundary nodes have least mutual correlation which can be denoted as* $\sum_{X_i,X_j} I(X_i,X_j)$ *described in Eq. 14 for all boundary nodes?* We need to compute $I(X_i,X_j)$ between two boundary nodes even before the probabilities are propagated and hence circuits features have to be used to measure mutual correlation between nodes. An easy index of correlation is nodes with same parent and grand parents are likely to be highly correlated.

Our aim is however to propose Tree-dependent distribution to capture the lost correlation during any partitioning. Hence, we heuristically tackled the largeness issue and boundary nodes correlation captured by TD. Steps involved in creating multiple Bayesian Networks from combinational circuits are as follows:
Step 1 Re-arranging the node numbering that such parents of a node are always numbered lower than the children. This ensures that all boundary nodes would have their ancestors in the previous BNs.

Step 2 Nodes with more than two parents are realized as a combination of nodes with two parents. These nodes are dummy nodes and as it can be seen from Figure 2 reduces maximum clique-size in (b) and hence reduces computational complexity. The accuracy of switching activity as depicted in Figure 2 is unaffected by this effort. Even for real-delay modeling, which we are currently investigating with Bayesian Networks, these dummy gates are assigned zero delay. While the TD construction, right now, we check all the boundary nodes for mutual correlation, the accuracy of tree-dependent distribution is not affected.

Step 3 Through experimentation, we figure the resource constraint of our computing device which is the allocated memory for running a single application. Our initial attempt is to take 500 nodes in one BN. This means that the first 500 nodes with all the edges are clubbed into one group. Maximum clique size is then estimated by the bounds described in previous section. If $\tau = 4^{C_{max}}.n$ where n is the number of nodes (and number of states for each node is 4) exceeds the resource constraint number, we reduce the last 50 nodes (and relevant edges) and try to infer the computational complexity and stop when τ roughly matches $4^{C_{max}}.n$. Thus, we get reasonably large individual networks. If τ is less than the resource constraint number we add 250 nodes with all the edges and re-estimate $4^{C_{max}}.n$ till it exceeds τ. The reduction is done then by 50 nodes as before till we get a match. The numbers (500, 250, 50) are obtained completely by trial and error. Thus, accuracy of the entire cascade structure is pre-dominantly determined by the internal nodes of the individual Bayesian Networks.

It has to be noted that we did not select BN by choosing least correlated boundary nodes at the boundary but in the next section, we discuss tree-dependent distribution and cascade structure which handles and re-capture lost information in the boundary nodes. In experimental results, we demonstrate the effect of TD on various partition points of c432 benchmark and show that TD improves the accuracy to a great extent when individual BNs are sufficiently large.

IV. Cascaded BN Modeling

In this section, we first introduce the Tree-Dependent distribution (TD) as an approximation for the actual probability function of the switching variables of interest. A Tree-Dependent distribution is the closest distribution to the actual underlying probability distribution of the switching variables with the constraint that the BN structure is a tree. This ensures that the TD is the best possible tree that captures the dependencies (spatial, temporal and spatio-temporal) closest to those exhibited by the actual distribution. Next, we utilize the approximate TD to model the correlations amongst the primary inputs as well as amongst the boundary nodes between the adjacent BNs in CBN model.

A. Tree-Dependent Distribution

We formally define a tree-dependent structure that we superimpose on the boundary nodes of the CBN's being cascaded. Next, we adapt the optimality results for TD function structure and hence, restate the proof of optimality from [3].

Definition 6: Any tree-dependent distribution $P^t(x)$ can be defined as a Markov field relative to the tree t which can be written as the product of $n-1$ pair-wise conditional probability distributions,

$$p^t(x) = \prod_i p(x_i|x_{j(i)}) \tag{11}$$

where $X_{j(i)}$ is the designated parent of X_i in some orientation of the tree t. The root node X_1 is chosen arbitrarily without any parents and $P(x_1|x_0) = P(x_1)$. Apart from the memory requirement, only second order statistics are needed to construct the tree.

Our goal is to construct a tree over n variables, representing the input nodes, that is the closest representation of the underlying joint probability function over the n variables. Hence, out of all the spanning tree over the n variables that can be constructed, we have to select the one which preserve the correlations to a maximum level. For this, we use a distance measure between two distribution P and $P^{'}$ known as Kullback-Leibler cross-entropy measure [4] in Eq. 12.

$$D(P,P^{'}) = \sum_{x} P(x)\log(P(x)/P^{'}(x)) \quad (12)$$

A low distance measure between P and P' indicates that the two distribution almost coincide with each other. Now, we have two subgoals: (1) To choose the best conditional probabilities between the parent and the child nodes in the tree given a fixed tree t such that P^t is the best approximation of P. This distribution is called the **projection of P on t, P_P^t** . And, (2) to choose a tree from a set of all the spanning trees over the nodes such that it would make the projection P on this tree P_P^t closest to P. We will use the two following theorems to arrive at a tree structure [3].

Theorem 2: The projection of P on t is characterized by the equality

$$P_P^t(x_i|x_{j(i)}) = P(x_i|x_{j(i)}) \quad (13)$$

Proof in [3].

This implies that the conditional probabilities for a branch a tree has to coincide with that computed from P will produce the best **projection of P on t, P_P^t**.

Theorem 3: The distance measure of Equation 12 is minimized by projecting P on any maximum weight spanning tree (MWST) where the weight of the branch (X_i, X_j) is defined by the information measure between them

$$I(X_i,X_j) = \sum_{x_i,x_j} P(x_i,x_j)\log\left(\frac{P(x_i,x_j)}{P(x_i)P(x_j)}\right) \quad (14)$$

Proof in [3].

We can use any algorithm for deriving the MWST. The steps we follow are listed below in Figure 3. Using a tree-structured representation ensures that storage proportional to $(r-1)r(n-1)+r-1$ [2] is used where r is the number of states (in our case r = 4) and n is the number of variables of the primary inputs which is much less than r^n, which would be needed for a complete representation. Moreover, by the above algorithms, we ensure that at least the pairwise correlations are captured effectively and propagated to the internal nodes.

B. Tree Dependent Input Modeling

The inputs to a combinational block can be correlated with each other. This may happen because the inputs could be outputs of another combinational block and its structural dependencies will force the primary inputs of the current blocks to be correlated. Moreover, high correlations are exhibited if the inputs are generated by some sequential circuits like counters. Switching correlations among the primary input nodes can affect switching activity estimates across the whole circuits. An ideal way to model the input switching would be, given a training set of input line transition, to learn an exact switching model in terms of a joint probability function over the input lines, which, of course, can also be represented by a Bayesian network (BN). This learned BN then would be coupled with the BN representation of the combinational circuit. There are, however, two practical problems associated with this approach.

1. Learning exact BN is NP-hard. Hence, in practice we can only learn an approximate BN representation.

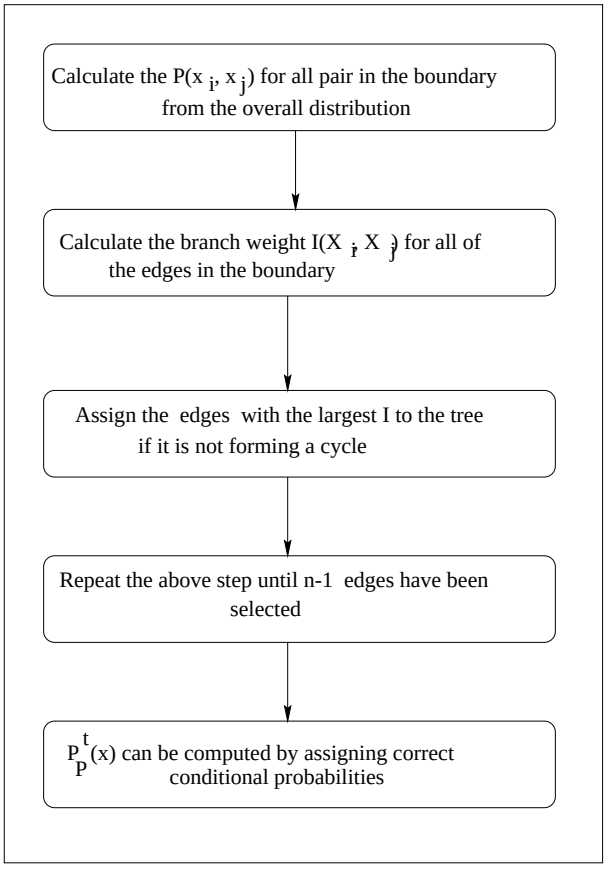


Fig. 3. Steps involved in construction of MWST

2. The exact BN representation might, depending on the correlations present amongst the input lines, have high computational complexity by itself (discussed in section III), thus allowing only a small sized CBNs and forcing a significant increase in the number of cascaded segments.

The large number of BNs will not only result in loss of accuracy, it will also increase the estimation time. Thus, we resort to TD based approximate modeling as a practical compromise of the accuracy of representation of dependencies and computational costs in terms of time and storage. For primary input modeling, we obtain the pairwise probability function based on the input statistics provided by the user, which we then use to compute the mutual information and the MWST, as shown as outlined in Figure 3.

C. Tree Dependent CBN Model

Since, large circuits with large *induced widths* can not be modeled using a single BN structure, partitioning into Cascaded BNs is inevitable. How do we couple these Cascaded-BNs? What kind of information passes between them? One possibility that we explore in [8] is naive coupling where two adjacent BNs have consistency in terms of *singleton* probabilities. However, this results in loss of accuracy for not capturing any correlation between the boundary nodes. One observation that helps is that some of the nodes of one BN are *inputs* to another BN. Thus, the BN coupling problem can be seen as an instance of the primary input modeling problem, where the “inputs” are not

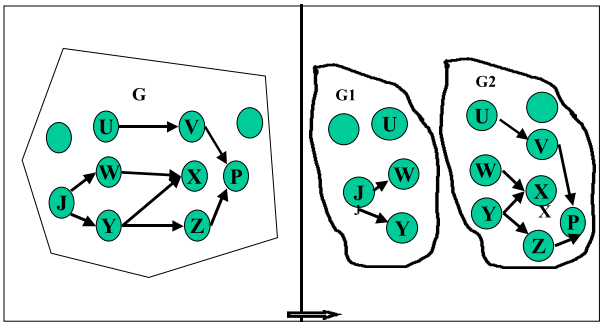


Fig. 4. Naively coupled adjacent BNs G_1 and G_2 of a graph G with common edges.

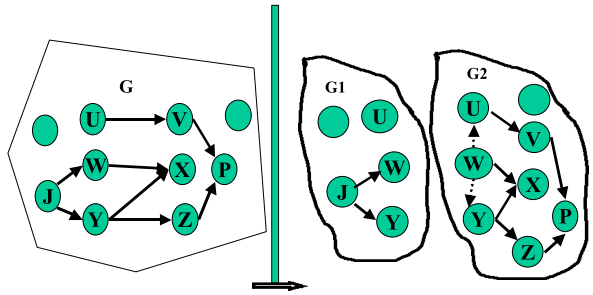


Fig. 5. TD coupled adjacent BNs G_1 and G_2 of a graph G with common edges.

the primary input lines but nodes in the previous BN. Hence, like the input modeling problem, we could use the approximate tree-dependent representation over the "input" nodes. Learning an accurate BN structure is, of course, another possibility, but that is computationally prohibitive. Besides, the accuracy achieved by an accurate Bayesian Network model over the segment boundaries may be nullified by the loss of accuracy due to increased number of BNs in the CBN that would be necessary. The accuracy improvement by accurate Bayesian Network modeling may become marginal as opposed to the computational time and memory requirements. Hence, as for primary input modeling, we resort to an approximation technique which is a good compromise between accuracy and memory requirement.

As an illustration of naive and TD-BN coupling, consider two segments G_1 and G_2, shown in Figure 4, which are two parts of one network G. Let the edges $(U,V),(W,X),(Y,X)$ and (Y,Z) be the four edges that have the first node in G_1 and the second node in G_2. In both naive and TD based coupling we remove edges $(U,V),(W,X),(Y,X)$ and (Y,Z) from G_1, however these common nodes U, W and Y are retained in G_1. We then reproduce the common nodes in G_2 along with the links $(U,V),(W,X),(Y,X)$ and (Y,Z) that were removed from G_1. In case of naive coupling (Figure 4), we treat these common nodes as primary inputs of G_2 and assign the singleton probabilities of these common nodes based on the computation of the inference engine from G_1.

However, in case of TD based CBN coupling, we do not want to lose the correlation between nodes U, W and Y in G_2. Hence we construct the tree-dependent distribution on these boundary nodes based on their mutual correlations in G_1. In G_2, we add all four edges, along with a tree structure between nodes U, W and Y shown in the dashed lines in Figure 5. This tree captures significant information from G_1 and propagates them to G_2. It has to be noted that tree-dependent distribution is computed purely by the information and the correlation exhibited between the boundary nodes in segment G_1. Since, any tree can be represented as a Bayesian Network, we cascade the tree into G_2 and obtain a combined Bayesian Network for the second segment. Thus, every Bayesian Network model of a part of the circuit now captures accurate correlation in the internal nodes of its own and captures significant correlations from the previous BN.

It has to be noted here that we will finally have $n-1$ edges, that would be selected from n^2 edges in each boundary. Moreover, computing mutual information between two nodes in different cliques is computationally expensive. The reason for that is the compiled BN itself complete use the existing resource and hence computing the marginal joint probabilities requires large memory accesses. In experimental results, we report our observation and it can be seen that building a tree without any heuristics, can consume large computational time which underplays the accuracy improvement. In the next section, we develop a heuristic in tree building that we use a good trade-off between accuracy and time.

D. Approximate CBN Modeling

In this subsection, we propose an approximate cascade structure which in theory is same as the CBN model discussed before and relies on the same tree-dependent distribution that is discussed in section A. However, we do not explore all the n^2 edges to compute the best $n-1$ edges. Instead, we use the circuit structure to select $n-1$ edges. We calculate mutual information between two boundary nodes only if they have a common child in the next BN segment or have common parent. Since our data structure stores parents of a node efficiently, we also calculate mutual information between nodes having common grandparents. In circuit terms, this implies that we collect mutual correlation for output signals of logic gates which have common inputs. Also, we calculate mutual information for signals that are inputs to one gate generating output signal. We present the steps to generate the heuristic tree for the tree-dependent distribution.

- Generate node pair in the boundary nodes that have common children.
- Generate node pair that have common inputs. This is performed before breaking larger fan-ins into smaller ones.
- Generate node pair that have common parents for parents of the boundary nodes. This is also performed in the original network.
- Calculate $P(X_i,X_j)$ for all the nodes from the above node pair i,j.
- Calculate $I(X_i,X_j)$ for all the nodes from the above node pair i,j.
- Assign the edges with largest I to the tree if it is not forming cycle.
- Repeat till $n-1$ edges are selected or if the all the node

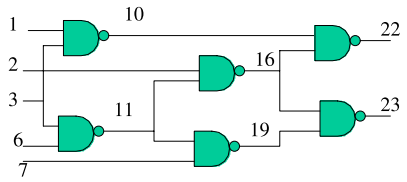


Fig. 6. Combinational circuit c17

pairs are considered.

- $P_P^t(x)$ is computed y assigning the conditional probability to the node pair.

V. Experimental Results and Conclusions

We mapped several ISCAS circuits to their corresponding Cascaded Bayesian Networks representation. The conditional probabilities are pre-determined by the type of gate connecting the parents and the child. We have already discussed in Section III that each node in Bayesian network represents switching at a line in the circuit and can be in one of the four states $(x_{00}, x_{01}, x_{10}, x_{11})$. Conditional probabilities are computed directly from logic structure of the gates. We used HUGIN's Bayesian Network tool [23] for compiling the junction tree and propagating the probabilities. We also performed in-house zero-delay logic simulation providing "ground truth" estimates of switching.

Using zero-delay model is a limitation for the current formalism. Most of the previous work in probabilistic modeling presents result on zero-delay simulation.

In our experiments, we want to establish that the tree-dependent distribution works accurately for input streams with different degree of correlation. Even for low input correlations the TD based Cascaded BN should perform better than BN modeling. It is an extremely essential feature of a power estimator to address inputs with a wide range of correlations. Many past estimators assumed random inputs. Marculescu et. al. experimentally modeled spatio-temporal dependencies in input correlations in [7]. The drastic effect of input correlations (low, medium and high) can be observed in Table IV.

First, let us consider the switching activity estimation of each node of the c17, which is a ISCAS benchmark that is small enough to be considered in this fashion under various degrees of correlations. The fan-out nodes from 3 are 8 and 9. Nodes 14 and 15 are fan-outs from 11 and nodes 20, 21 are fan-outs from 16. As it can be seen in Figure 6, that the nodes 10 and 11 are highly correlated and the nodes 16 and 19 are highly correlated to each other. Since the nodes 10 and 16 both are dependent on node 3 they are correlated too. We performed three experiments (i) inputs with low correlations (random inputs), (ii) temporally biased inputs and (iii) highly correlated counter generated inputs. Table II lists the estimates for all the nodes. As it can be observed, at all the nodes, we have extremely accurate estimates. Moreover, it becomes very evident that the average switching activity for these three cases are indeed affected by the dependencies in the inputs (0.462 for random inputs, 0.347 for temporally biased inputs and 0.297 for highly correlated inputs) from each other showing us the need for input-modeling.

TABLE II
Comparison of estimated switching activity by BN modeling and simulated switching activity of each node of benchmark c17 for different input types.

Node	Random Inputs		Temporally Correlated		Spatio-temporally Correlated	
	Est. Sw_{BN}	Sim. Sw	Est. Sw_{BN}	Sim. Sw	Est. Sw_{BN}	Sim. Sw
1	0.5	0.499	0.4	0.401	0.063	0.063
2	0.5	0.5	0.4	0.402	0.125	0.125
3	0.5	0.499	0.4	0.4	0.25	0.25
6	0.5	0.5	0.4	0.4	0.5	0.5
7	0.5	0.5	0.4	0.4	1.0	1.0
8	0.5	0.499	0.4	0.4	0.25	0.25
9	0.5	0.499	0.4	0.4	0.25	0.25
10	0.375	0.375	0.16	0.16	0.125	0.125
11	0.375	0.374	0.16	0.16	0.25	0.25
14	0.375	0.374	0.16	0.16	0.25	0.25
15	0.375	0.374	0.16	0.16	0.25	0.25
16	0.469	0.469	0.38	0.382	0.125	0.125
19	0.469	0.469	0.38	0.379	0.75	0.75
20	0.469	0.469	0.38	0.382	0.125	0.125
21	0.469	0.469	0.38	0.382	0.125	0.125
22	0.492	0.493	0.435	0.436	0.125	0.125
23	0.492	0.492	0.48	0.481	0.5	0.5

It is worth noting that partition points would have impact on accuracy and in this paper, our partition is based on resource-constraints. The partitioning algorithms that reduces correlated boundary nodes, would enhance accuracy of this model further. Table III shows the BN and approximate CBN model for different partition points for benchmark circuit c432. It is clear that CBN offers lower mean and deviation for almost all cases over naive BN.

Next, we show results on other benchmark circuits. As before, we use a random number generator (rand function in C++) for the inputs exhibiting low correlations (Table V). Adopting Marculescu et al.'s [7] strategy, we generated a medium correlation scenario by sorting the sequences from the random number generator (Table VI). It has to be the noted that by sorting the signal probabilities of each input node remains the same as that produced by the random generator but correlations between the nodes are introduced. Finally, we use 16 bit counters to generate highly correlated sequences for the experiments (Table VII). These errors and standard deviation of errors (RMS) are obtained by comparing the estimated value of switching activity by CBN model and the switching activity values obtained from an in-house zero-delay logic simulators.

In Table V, we show three types of Bayesian Network based model for inputs with low correlation. The first set denotes naive Bayesian Network coupling. This model makes all the

TABLE V

COMPARISON OF RESULTS BETWEEN BN AND CBN BASED MODEL FOR INPUTS WITH LOW CORRELATIONS

	BN model				CBN model					Approx. CBN model				
circuit	μ	σ	max	Time(s)	μ	σ	max	T-BN(s)	T-TD(s)	μ	σ	max	T-BN(s)	T-Th(s)
c432	0.006	0.031	0.29	0.16	0.002	0.032	0.197	0.36	164.58	0.003	0.023	0.287	0.171	2.83
c499	0	0.004	0.0234	0.08	0	0.001	0.006	0.09	8.94	0	0.001	0.008	1.14	7.49
c880	0.001	0.009	0.066	0.61	0.001	0.009	0.066	0.71	0.26	0	0.004	0.04	0.44	2.08
c1355	0.006	0.033	0.188	0.4	0.001	0.007	0.124	0.491	15.61	0	0.009	0.09	0.5	1.31
c1908	0.001	0.01	0.155	1.0	0.001	0.01	0.099	1.28	988.68	0.001	0.01	0.15	1.36	9.34
c3540	0.003	0.044	0.279	4.4	0.005	0.037	0.252	4.72	3411.7	0	0.041	0.26	7.31	11.55
c6288	0.014	0.046	0.421	4.5	0.006	0.023	0.313	4.857	531.26	0.013	0.041	0.375	17.26	21.49

TABLE VI

COMPARISON OF RESULTS BETWEEN BN AND CBN BASED MODEL FOR INPUTS WITH MEDIUM CORRELATIONS

	BN model				Approx. CBN model				
Circuits	μ	σ	*Max*	Time(s)	μ	σ	*Max*	$T_{BN}(s)$	T-tree(s)
c432	0.004	0.042	0.171	0.22	0.000	0.020	0.183	0.16	0.94
c499	0.004	0.021	0.190	0.11	0.001	0.002	0.020	0.12	1.1
c880	0.004	0.0269	0.278	0.14	0.004	0.023	0.278	0.16	0.04
c1355	0.009	0.037	0.194	0.30	0.010	0.035	0.189	0.38	2.57
c1908	0.016	0.038	0.192	0.95	0.016	0.028	0.191	0.96	2.13
c3540	0.001	0.050	0.298	1.92	0.001	0.042	0.257	3.10	2.81
c6288	0.016	0.066	0.300	2.30	0.010	0.055	0.293	5.39	5.16

TABLE VII

COMPARISON OF RESULTS BETWEEN BN AND CBN BASED MODEL FOR INPUTS WITH HIGH CORRELATIONS.

	BN model				Approx. CBN model				
Circuits	μ	σ	*Max*	Time(s)	μ	σ	*Max*	$T_{BN}(s)$	T-tree(s)
c17					0.000	0.000	0.00	0.000	0.001
c432	0.003	0.027	0.192	0.07	0.003	0.020	0.141	0.61	45.49
c499	0.002	0.023	0.172	0.09	0.000	0.018	0.113	0.12	2.10
c880	0.000	0.002	0.033	0.14	0.000	0.002	0.031	0.19	0.07
c1355	0.007	0.039	0.191	0.37	0.001	0.017	0.191	0.70	0.72
c1908	0.006	0.027	0.388	0.95	0.005	0.027	0.388	1.11	54.08
c3540	0.000	0.001	0.083	2.6	0.000	0.001	0.083	3.26	0.80
c6288	0.009	0.034	0.450	2.7	0.006	0.027	0.440	3.09	12.58

Bayesian Networks consistent on the singleton probabilities of the boundary nodes. The mean, standard deviation, maximum error and total computation time is reported in column 2, 3, 4 and 5. The next set is based on tree-dependent CBN model. In this set, we explore all the n^2 possible edges in the boundary nodes to arrive at the best Maximum weight spanning tree. The mean, standard deviation, maximum error and total computation time in Bayesian Networks and in the tree computation is reported in column 6, 7, 8, 9, and 10 respectively. Accuracy of this model is much higher than that of the naive coupling. It is obvious that the time to build the tree can be high for some benchmarks. In the third set, we report results from the approximate CBN model where the best tree is selected from the nodes which have common inputs, or which are common inputs for another gate (discussed in section D. The mean, standard deviation, maximum error and total computation time in Bayesian Networks and in the tree computation is reported in column 11, 12, 13, 14, and 15 respectively. It is obvious that these tree-dependent approximate model is the best trade-off between accuracy and speed.

For medium and high correlation scenarios, we report and compare results for the naive coupled BN and the approximate CBN models. It can be observed that the TD-CBN results in high accuracy in terms of mean and standard deviation of error. With TD, the mean is in general lower than that without TD-based models. Standard deviation of error which signifies

TABLE III
MODELING ERRORS FOR TEN DIFFERENT PARTITIONS OF C432.

	BN model			CBN model with TD		
#	μ	σ	*Max*	μ	σ	*Max*
3	0.005	0.054	0.228	0.004	0.052	0.238
4	0.008	0.059	0.208	0.004	0.053	0.237
5	0.010	0.061	0.203	0.010	0.061	0.207
3	0.007	0.034	0.250	0.003	0.024	0.287
5	0.009	0.060	0.208	0.005	0.052	0.229
7	0.007	0.059	0.219	0.005	0.052	0.259
4	0.008	0.057	0.223	0.005	0.054	0.256
6	0.008	0.058	0.223	0.008	0.057	0.244
3	0.007	0.034	0.255	0.002	0.025	0.303

TABLE IV
SWITCHING ACTIVITY AND PRIMARY INPUT CORRELATION

	Average Switching Activity under		
Circuits	Low Corr.	Medium Corr.	High Corr.
c432	0.39	0.24	0.04
c499	0.42	0.35	0.19
c880	0.39	0.29	0.03
c1355	0.36	0.29	0.10
c1908	0.41	0.30	0.10
c3540	0.37	0.33	0.00
c6288	0.42	0.33	0.01

the diversity of error estimates are 1.5 to 2 times smaller with TD based coupling than the naive one for most cases. For many circuits, we have significant reduction of the maximum errors. The reduction in standard deviation and the maximum errors signifies that the estimation based on CBN models are not only more accurate but also more uniform. The error distribution with CBN and BN models for both low, medium and highly correlated inputs for benchmark c1355 and c6288 are shown in Figures 7, 8 (CBN in yellow (or light) and BN in red (or dark)). We exclude all the nodes that are estimated close to zero error to have a fair comparison of high-end errors. CBN has significantly higher number of nodes in the zero-error range (though not seen in the graph) which is exactly the expected behavior. It is obvious that with TD based CBN modeling, we achieve very low error spread compared to naive BN models. We used a DELL PC with WINDOWS (98 SE) operating system running at 2 GHz and the maximum time for the input modeling was 1.32*s*. We can conclude that TD based CBN modeling is essential for accurate and uniform switching activity estimation of all the nodes in the entire circuit and is also equally important for modeling correlated inputs. We compare our work with a few existing work in Table VIII for random inputs and show that we perform really well in terms of accuracy of estimation.

VI. CONCLUSION

This paper advances the accuracy of the Bayesian network based switching activity estimation tool, which captures dependencies, both in the internal nodes and in inputs, in reasonable time and with high accuracies. We suggest ways to handle large circuits using Cascaded BNs where each BN in the cascade exactly models spatio-temporal dependencies among the nodes and approximate couplings between BNs are performed by TD distributions, which are a compromise between space complexity and accuracy. This approach is stimulus-free even for modeling correlated input streams. We also discuss complexity issues for BN inference schemes and suggest partitioning guidelines to form the CBN. The results are very competitive in terms of accuracy and the elapsed time for estimation. Our future effort will focus on modeling delays.

TABLE VIII
COMPARISON WITH SCHEIDER *et al.* [19] AND MARCULESCU *et al.* [7].

	Ref. [19]	Ref. [7]		CBN-TD	
Circ.	μ_{Err}	μ_{Err}	*Max*	μ_{Err}	*Max*
c432	0.016	0.028	0.210	0.002	0.197
c499	-	0.013	0.062	0	0.006
c880	0.006	0.013	0.069	0.001	0.066
c1355	0.005	0.004	0.003	0.001	0.124
c1908	0.010	0.009	0.131	0.001	0.099
c3540	0.014	0.030	0.201	0.005	0.252
c6288	0.023	0.014	0.089	0.006	0.318

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

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

[5] 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.

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

[7] 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.

[8] S. Bhanja, N. Ranganathan, "Dependency Preserving Probabilistic Modeling of Switching Activity using Bayesian Networks", *38th Design Automation Conference*, pp. 209–214, 2001.

[9] S. Bhanja, N. Ranganathan, "Swtiching activity estimation of VLSI circuits using Bayesian networks", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11-1, pp. 558–567, 2003.

[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 Computer-Aided Design of Integrated Circuits and Systems*, vol 12-2, pp. 310–323, Feb. 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. 794–800, June 1992.

[13] 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. 17–11, pp. 1099–1107, November, 1998.

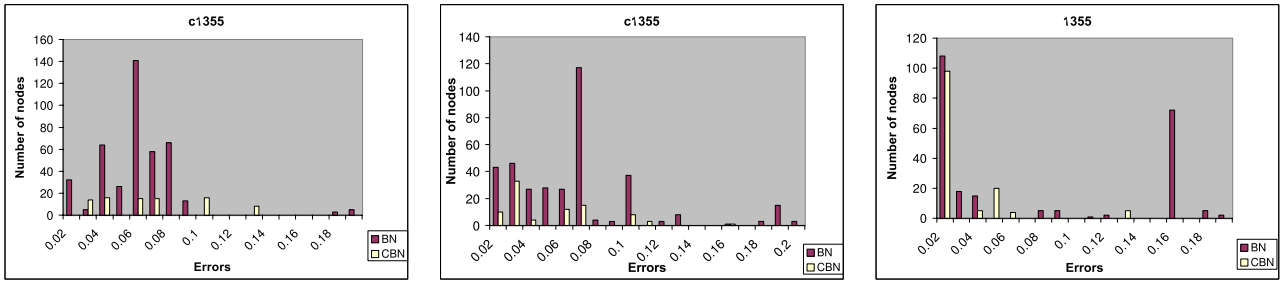


Fig. 7. Error distribution for c1355 for input streams with low (a), medium (b) and high (c) correlation. (BN: dark or red; CBN: light or yellow)

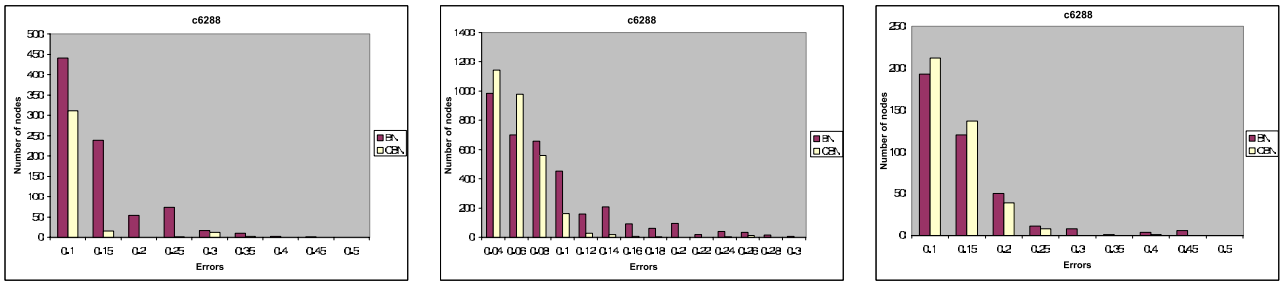


Fig. 8. Error distribution for c6288 for input streams with low (a), medium (b) and high (c) correlation. (BN: dark or red; CBN: light or yellow)

[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. ACM/IEEE Design Automation Conference*, pp. 224–228, Nov. 1993.

[17] 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. 9-4, 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.

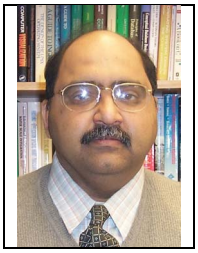
[20] A. K. Murugavel and N. Ranganathan, " Petri net modeling of gate and interconnect delays for power estimation ", *Design Automation Conference*, pp. 455–460, 2002.

[21] R. Marculescu, D. Marculescu, and M. Pedram, "Switching Activity Analysis Considering Spatiotemporal Correlations", *Proc. 1994 Intl. Conference on Computer Aided Design*, pp. 294-299, Nov. 1994.

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

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

Sanjukta Bhanja received her BE and MSc (Engg.) in Electrical Engineering from Jadavpur University, Calcutta and the Indian Institute of Science, Bangalore, respectively. She received her Ph.D in Computer Science and Engineering from University of South Florida, Tampa in Dec. 2002. She is currently an Assistant Professor in the Department of Electrical Engineering at the University of South Florida, Tampa. Her research interests includes design automation, low power VLSI design, power estimation, optimization, system-on-a-chip, graphical probabilistic belief model, dependency analysis, computing under uncertainty in devices, VLSI testing, and expert systems based modeling.



Nagarajan (Ranga) Ranganathan (S'81-M'88-SM'92, F'02) received the B.E. (Honors) degree in electrical and electronics Engineering from Regional Engineering College, Tiruchirapalli, University of Madras, India in 1983, and the Ph.D. degree in computer science from the University of Central Florida, Orlando in 1988.

He is currently a professor in the Department of Computer Science and Engineering and the Nanomaterials and Nanoelectronics Research Center at the University of South Florida, Tampa. His research interests include VLSI system design, VLSI design automation, low power design, computer architecture and bioinformatics. He has developed many special purpose VLSI chips for computer vision, image processing, pattern recognition, data compression and signal processing applications. He has co-authored about 200 papers in reputed journals and conferences and is a co-owner of five U.S. patents. He was elected as Fellow of IEEE for his contributions to algorithms and architectures for VLSI systems design.

Dr. Ranganathan is a member of IEEE, IEEE Computer Society, IEEE Circuits and Systems Society and the VLSI Society of India. He served as the chair of the IEEE Computer Society Technical Committee on VLSI during 1997-2000. He has served on the program committees of international conferences such as ISLPED, ICCD, CAMP, ICPP, IPPS, SPDP, VLSI Design and ICHPC. He has served on the editorial boards of various journals such as Pattern Recognition, Intl. Journal of VLSI Design, IEEE Transactions on VLSI Systems, IEEE Transactions on Circuits and Systems TCAS-II, and IEEE Transactions on CAS for Video Technology. He served as the steering committee chair for the IEEE Transactions on VLSI Systems during 2000-02. He is currently serving as the Editor-In-Chief of the IEEE Transactions on VLSI Systems. He received the USF Division of Sponsored Research Outstanding Research Achievement Award in 2002, the USF President's Faculty Excellence Award in 2003, the Theodore-Venette Askounes Ashford Distinguished Scholar Award in 2003, and the SIGMA XI Scientific Honor Society Tampa Bay Chapter Outstanding Faculty Researcher Award in 2004. He was a co-recipient of two Best Paper Awards at the Intl. Conf. on VLSI Design in 1995 and 2004.