

A Stimulus-free Graphical Probabilistic Switching Model for Sequential Circuits using Dynamic Bayesian Networks

SANJUKTA BHANJA*, KARTHIKEYAN LINGASUBRAMANIAN* and N. RANGANATHAN[†]

* Department of Electrical Engineering

[†] Department of Computer Science and Engineering

University of South Florida, Tampa, FL

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

Categories and Subject Descriptors: B.8.2 [Performance Analysis and Design Aids]: Power estimation

This work is partially sponsored by University of South Florida through the New Researcher Grant.

Start of a second footnote ...

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20 ACM 1084-4309/20/0400-0001 \$5.00

1. INTRODUCTION

The ability to form accurate estimates of power usage, both dynamic and static, of VLSI circuits is an important issue for rapid design-space exploration. In circuits that are in active mode most of the time or those that switch between the active and standby modes, the total power (both static and active) becomes strongly input dependent [Nguyen et al. 2003; Srivastava et al. 2004]. In this work, we capture this input dependence by constructing a switching model.

Note that the switching model is extremely relevant of both static and dynamic component of power as shown in Eqn. 1 [Nguyen et al. 2003]. In this equation, P_{dg} represents the dynamic component of power at the output of a gate g . The impact of data on dynamic component of power is encapsulated in α , the individual switching activity. The static component of power P_{sg} is dominated by $P_{leak,i}$, leakage loss in a leakage mode i . It has to be noted that each leakage mode is determined by the steady state signals that each transistor in the gate would be in. For example, in a two input (say A and B) NAND gate, the gate would have four dominant leakage mode ($i=4$: $A@_0B@_0, A@_0B@_1, A@_1B@_0$ and $A@_1B@_1$). β is the probability of each mode i and for example $\beta_1 = p(A@_0, B@_1)$ and is the joint probability of multiple signals in a gate and are dependent on the input data profile.

$$\begin{aligned} P_t &= \sum_g P_{tg} = P_{dg} + P_{sg} \\ &= 0.5\alpha f V_{dd}^2 C_{load+wire} + \sum_i P_{leak,i} \beta_i \end{aligned} \quad (1)$$

Switching activity α in Eq. 1 is one of the important component that is a direct yield from the switching model and contributes to the dynamic power dissipation that is independent

of the technology of the implementation of the VLSI circuit. Contribution to total power due to switching is dependent on the logic of the circuit and the inputs and will be present even if sizes of circuits reduce to nano domain. Apart from contributing to power, switching in circuits is also important from reliability point of view and hence can be considered to be fundamental in capturing the dynamic aspects of VLSI circuits.

Among different types of VLSI circuits, switching in sequential circuits, which also happens to be the most common type of logic, is the hardest to estimate. Even though a large set of research work is performed in this area, almost all of it are input stimuli based and focus of the research is centered around obtaining a correct set of prior probabilities for the state nodes. This is particularly due to the complex higher order dependencies in the switching profile, induced by the spatio-temporal components of the main circuit but mainly caused by the state line feedbacks that are present. These state line feedbacks are not present in pure combinational circuits. One important aspect of switching dependencies in sequential circuits that one can exploit is the first order Markov property, i.e. the system state is independent of all past states given just the previous state. This is true because the dependencies are ultimately created due to logic and re-convergence, using just the current and last values.

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

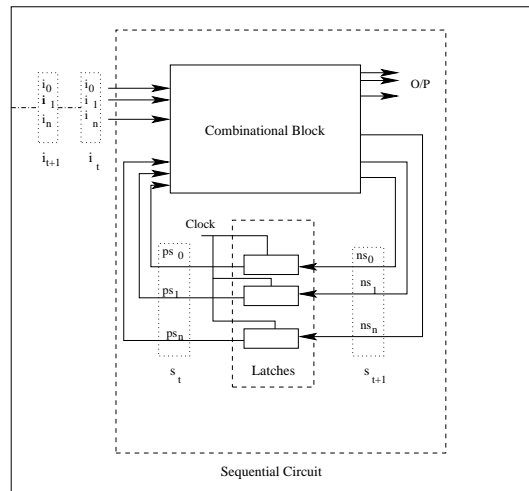


Fig. 1. A model for sequential circuit.

and temporal correlations at the state lines are of great interest. It is important to be able to model both kinds of dependencies in these lines.

Previous work in [Bhanja and Ranganathan 2001] has shown that a combinational circuit can be *exactly* modeled in a probabilistic Bayesian Network (BN). Such models capture both the temporal and spatial dependencies in a compact manner using conditional probability specifications. For combinational circuits, first order temporal models are sufficient to completely capture dependencies under zero-delay. The attractive feature of the graphical representation of the joint probability distribution is that not only does it make the conditional dependencies among the nodes explicit, but it also serves as a computational mechanism for efficient probabilistic updating. This BN structure, however, cannot model cyclical logical structure, like those induced by the feedback lines. This cyclic dependence affects the state line probabilities that, in turn, affects the switching probabilities in the entire circuit.

1.1 Overview

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

We consider two inference algorithms to form the estimates from the built TDM representations: one is an exact scheme and the other is a hybrid scheme. The exact inferences scheme, which is based on local message passing, is presently practical for small circuits due to computational demands on memory. For large circuits, we resort to a hybrid inference method based on a combination of local message passing and importance sampling. Note that this sampling based probabilistic inference is non-simulative and is different from

samplings that are commonly used in circuit simulations. In the latter, the input space is sampled, whereas in our case both the input and the line state spaces are sampled simultaneously, using a strong correlative model, as captured by the Bayesian network. Due to this, convergence is faster and the inference strategy is stimulus-free.

The key features of this work are:

(1) This is a completely probabilistic model. This means that we do not perform input-pattern driven simulation rather focus on a random walk in the probabilistic dependency model. Hence we target the “circuit problem” rather than the “input problem” [Marculescu et al. 1999].

(2) In this work we introduce a method for handling higher order temporal dependencies in sequential circuits using dynamic bayesian networks.

(3) This work use a stochastic and accurate inference schemes for propagating probabilities in the dynamic Bayesian Networks.

(4) Since we represent the joint probability distribution of the entire variables including switching of state and signals, our model is capable of accurately predicting the coupling behavior of not only the individual switching activities, but also the probability of joint occurrence of events namely probability of two neighboring signals in a transistor stack remaining at *zero*. We also get $P(X_{0 \rightarrow 1}, Y_{1 \rightarrow 0})$ for two neighboring lines that measures how often these two signals would be in the worst case cross-talk situation.

(5) These probabilistic set up are extremely capable and are mostly used for non-causal backward inferencing along with the forward propagation. For example, we would be able to see what type of input profile provides a low switching for a node that has high load capacitance and these could be used to characterize the input space. With probabilistic forward propagation, we need to an evolutionary search to find such characterization.

2. PRIOR WORK

Since our work falls in the broad category of probabilistic methods of estimation, we start with a few break-throughs in probabilistic estimation for combinational circuit. In [Najm 1993], 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 [Bryant 1992] which has a high space requirement. Tagged probability simulation is proposed in [Ding et al. 1998], which is based on the local OBDD(ordered binary decision diagram) propagation. The signal correlations are captured by using local OBDDs. However, spatio-temporal correlation between the signals is not discussed. Pair-wise correlation between circuit lines were first proposed by Ercolani *et al.* in [Ercolani et al. 1992]. Marculescu *et al.* in [Marculescu et al. 1998], studied temporal, spatial and spatio-temporal dependencies as a composition of pair-wise correlations. However, none of these methods could be extended for modeling the dynamics of feedback system.

In one of a pioneering work, Marculescu et al. [Marculescu et al. 2000] proposed a method to find upper and lower bounds for the switching activity in finite state machines (FSMs), using a Markov chain model. This method even though probabilistic is more appropriate when circuit implementation of FSM is not known. In another approach Marculescu et al. used dynamic Markov model to capture the probabilistic temporal structure of the input space in sequential circuits. [Marculescu et al. 1999]. Once a reduced length input data-set is obtained, this method relies on simulation of the circuit for computing the power. In their own words, this work is focused to solve the “input problem” [Marculescu et al. 1999] rather than modeling the circuit probabilistically.

Bhanja *et al.* [Bhanja and Ranganathan 2001], handle the underlying joint probability distribution for the combinational circuit. In [Bhanja and Ranganathan 2004], we have

also modeled the probabilistic framework for input space by learning a best-fit tree structured Bayesian Networks in inputs which was also used in the partitions of intermediate Bayesian Networks. Hence both of these models were not capable of handling dynamic temporal aspects of a sequential circuits. In [Bhanja and Ranganathan 2004] we focused on mainly three aspects (1)issue of complexity and partitioning heuristics (2) Capturing correlation at the boundaries of adjacent loosely coupled Bayesian Networks and (3) a tree structure learning algorithm for spatio-temporally coupled (correlated) inputs. Sequential circuits have a different problem. Even with the random (spatio-temporally independent) inputs, the spatio-temporal correlation is generated in the circuit due to the feedback mechanism and this increases an induced dependency in the inputs which needs to be modeled separately along with the state feedback.

Most existing techniques for switching estimation in sequential circuits use statistical simulation. Almost all the statistical techniques in one way or the other employ sequential sampling of inputs, along with stopping criteria determined by the assumed statistical model. Stamoulis *et al.* [Stamoulis 1996] considered a path oriented transition probability computation to sample the input signal space, but the model did not account for correlations between latch and the combinational part. Najm *et al.* [Najm et al. 1995] proposed logic simulation to obtain the state line probability and then used statistical simulation for estimation.

Tsui *et al.* [Tsui et al. 1995] used a two-step approach. First, the stationary state line probabilities were estimated instead of state probabilities, thereby reducing the problem complexity but sacrificing the ability to model the spatial dependencies among the state lines. A set of non-linear equations described the relations of the next state to the primary inputs and the current state lines. These equations are solved using the Picard-Peano and Newton-Raphson methods to arrive at locally optimal solution for the individual state line probabilities. Next, given these state probability estimates, statistical simulations (or,

ACM Transactions on Design Automation of Electronic Systems, Vol. , No. , 20.

for very small circuits, binary decision diagrams (BDD)) were used to obtain switching estimates. Yuan *et al.* [Yuan et al. 1997] exploited the observation that beyond a length of samples the inputs can be considered independent. Saxena *et al.* [Saxena et al. 2002] simulated multiple copies of a circuit, with mutually independent input vectors, thereby generating mutually independent samples. The entire sequential circuit estimates were formed by Monte-Carlo simulation with these inputs. Chen *et al.* [Chen and Roy 1997] presented a technique to estimate upper and lower bounds of power by considering the signal probability and signal activity at the inputs. Kozhaya *et al.* [Kozhaya and Najm 2001] enhanced this method in their work to estimate power in sequential circuits.

Even the best simulation based methods suffer from weak input pattern dependencies. Besides, for any change in primary input statistics, simulations need to be rerun. For these reasons, we prefer probabilistic strategies, particularly those that model the underlying joint probability distribution of the node switching. Such probabilistic models, not only allows one to estimate the switching probabilities, but also readily facilitates conditional estimation, i.e. estimation conditioned on the knowledge of the switching statistics at some nodes, not necessarily the input lines. In this work, we propose a graphical probabilistic model over all the state nodes, inputs, and internal lines to accurately model the joint switching probabilities of the whole circuit. The model accounts for high order temporal and spatial dependencies. The exact spatio-temporal correlation is modeled over n time slices to capture higher (> 1) order temporal effects that are present in sequential circuits. Circuits modeled using n slices captures n -th order temporal dependencies. Similar strategy was used by Tsui *et al.* [Tsui et al. 1995], who “unrolled” the circuits n times to arrive at the statistics of the individual state lines, which were then used in simulations. We do not restrict ourselves to just state lines. Our comprehensive probability model, spanning multiple time slices, is over all the lines and simultaneously estimates the switching probabilities at all the lines (primary input, internal, and state lines).

3. BACKGROUND ON DYNAMIC BAYESIAN NETWORKS

In this section, we discuss the structure and fundamentals of dynamic Bayesian Networks, which underlies our modeling of sequential circuits as TDMs. Since Dynamic Bayesian Networks are structurally Bayesian Networks themselves, we will highlight important features of Bayesian Networks as well. The following discussion is fashioned after [Pearl 1988; Kjaerulff 1995]. As a peek ahead, the reader can look at Fig. 2 for an example of the representation for a small circuit.

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

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

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

$$P(x_1, \dots, x_N) = \prod_v P(x_v | Pa(X_v)) \quad (3)$$

where $Pa(X_v)$ are the parents of the variable x_v , representing its direct causes. For example in Fig. 2a X_1, X_2 are the parents of X_3 , X_2 is the parent of X_4 . This factoring of the joint probability function can be represented as a directed acyclic graph (DAG), with nodes (V) representing the random variables and directed links (E) from the parents to the children, denoting direct dependencies. The size of the representation, and hence the computational complexity would be dependent on the number of parents per node.

This section discusses some fundamental concepts behind the Bayesian Network based modeling. This is fashioned after [Bhanja and Ranganathan 2003] and is re-iterated here

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

for the clarity of the readers. The important aspects that signifies a directed acyclic graph as a Bayesian Network are *conditional independence* and *d-separation*. The DAG structure preserves all the *independencies* among sets of random variables and is referred to as a Bayesian network. The concept of Bayesian network can be precisely stated by defining the notion of *conditional independence* among a set of random variables.

Definition 1: Let $U = \{\alpha, \beta, \dots\}$ be a finite set of variables taking on discrete values. Let $P(\cdot)$ 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

$$P(x|y,z) = P(x|z) \text{ whenever } P(y,z) > 0 \quad (4)$$

Following Pearl [Pearl 1988], 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 . For example in Fig. 2a (without the feedback line), we have $I(X_6, X_3, X_1)$ which denotes X_6 and X_1 are conditionally independent given X_3 .

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 , $\langle X|Z|Y \rangle$, 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 .

Along with the above definitions the following definitions for conditional independence map or *I-map* and *minimal I-map* justifies the acceptance of a DAG as a BN.

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 nodes X , Y and Z we have, $\langle X|Z|Y \rangle \Rightarrow I(X, Z, Y)$.

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

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 $U = \{X_1, \dots, X_n\}$ of elements, and let d be an ordering $\{X_{d1}, X_{d2}, \dots\}$ of the elements of U . The *boundary strata* of M relative to d is an ordered set of subsets of U , $\{B_{d1}, B_{d2}, \dots\}$ such that each B_i is a Markov boundary (defined above) of X_{di} with respect to the set $U_i (\subset U) = \{X_{d1}, X_{d2}, \dots, X_{d(i-1)}\}$, 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 [Pearl 1988]. This theorem is the key to constructing a Bayesian network over multiple time slices (Dynamic Bayesian Networks).

Theorem 1: If a graph structure D is a boundary DAG of a dependency model M relative to ordering d , then D is a minimal I-map of M .

Proof: Please refer to [Pearl 1988].

3.1 DBN Structure

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

$$V = \bigcup_{i=1}^n V_{t_i} \quad (5)$$

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

$$E_{t_i, t_{i+1}} = \{(X_{i, t_i}, X_{j, t_{i+1}}) | X_{i, t_i} \in V_{t_i}, X_{j, t_{i+1}} \in V_{t_{i+1}}\} \quad (6)$$

where X_{j, t_k} is the j -th node of the DAG for time slice t_k . Note that in a generalized structure (Fig. 2b), temporal edges can be drawn from any node of the time slice t_k to any node of the time slice t_{k+1} . However, the overall representation has to be the minimal I-map of the underlying probabilistic model (Fig. 2c).

Thus the complete set of edges E is

$$E = E_{t_1} \cup \bigcup_{i=2}^n (E(t_i) + E_{t_{i-1}, t_i}) \quad (7)$$

Apart from the independencies among the variables from one time slice, we also have the following independence map over variables across time slices if we assume that the random variables representing the nodes follow Markov property [Hachtel et al. 1994], which is true for switching.

$$I(\{X_{j, t_1}, \dots, X_{j, t_{i-1}}\}, X_{j, t_i}, \{X_{j, t_{i+1}}, \dots, X_{i, t_{i+k}}\}) \text{ is true } \forall i > 1, k > 1 \quad (8)$$

where $I(X,Z,Y)$ implies conditional independence.

4. TDM MODELING

The core idea is to express the switching activity of a circuit as a joint probability function, which can be mapped one-to-one onto a Bayesian Network, while preserving the dependencies. To model switching at a line, we use a random variable, X , with four possible states indicating the transitions from $\{x_{00}, x_{01}, x_{10}, x_{11}\}$. For combinational circuits, directed edges are drawn from the random variables representing switching of each gate input to the random variable for switching at the outputs of that gate. At each node, we also have conditional probabilities, given the states of parent nodes. If the DAG structure follows the logic structure then it is guaranteed to map all the dependencies inherent in the *combinational* circuit. However, sequential circuits cannot be handled in this manner. We have proposed a switching activity estimation model for combinational circuits. A straightforward extension of this model would result in a graph structure shown in Figure 2(a), which is not a DAG and hence is not a Bayesian network.

4.1 Structure

Let us consider graph structure of a small sequential circuit shown in Fig. 2(a). Following logic structure will not result in a DAG; there will be directed cycles due to feedback lines. To handle this, we do not represent the switching at a line as a single random variable, X_k , but rather as a set of random variables, representing the switching at consecutive time instants, $\{X_{k,t_1}, \dots, X_{k,t_n}\}$, and then model the logical dependencies between them by two types of directed links.

(1) For any time instant, edges are constructed between nodes that are logically connected in the combinational part of the circuit, i.e. without the feedback component. Edges are drawn from each random variable representing switching activity at each input of a gate to the random variable representing output switching of the gate.

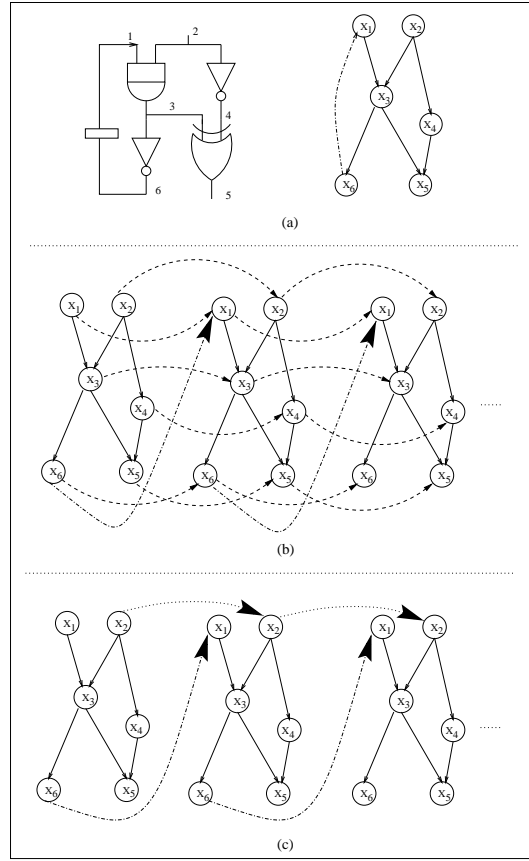


Fig. 2. Comparison of possible graphical models of a sequential circuit (left of part (a)). The representations shown in right side of (a) and in (b) are not used. (a) If one were to treat the circuit as a combinational circuit one gets a logically induced graph structure [Bhanja and Ranganathan 2001]. (b) Time unrolled representation. (c) TDM representation that we propose. Nodes in the graphs represent line switchings.

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

be $0 \rightarrow 0$.

Let us consider Figure 2(b), note that we have all random variables connected to the previous time slice for every slice. This is required as our random variables are switching activities and not logic. Hence, every random variable $X_{i,t_{i-1}}$ in this experiment has an edge connecting the variable X_{i,t_i} . However, this representation is *not* a DBN as it is not minimal. Markov blanket of the switching variables at a time instant t_i are still its parents and hence for individual node X_{i,t_i} , $I(X_{i,t_i}, Pa(X_{i,t_i}), X_{i,t_{i-1}})$ is true, and this implies that given the set of parents $Pa(X_{i,t_i})$, X_{i,t_i} is independent of $X_{i,t_{i-1}}$. Hence we could remove edges from $X_{i,t_{i-1}}$ to X_{i,t_i} without hampering the dependency model. However, for primary inputs, we have to make sure that $P(X_{k,t_{i+1}}^p | X_{k,t_i}^p)$ handles the switching constraints properly. In sequential circuit, each input signal values are considered random, however, for modeling the switching variables the temporal edges in inputs are essential and eliminates the need for explicit temporal dependencies of the intermediate random variables.

We call this graph structure as the temporal dependency model or TDM. Fig. 2(c) shows the TDM for the example sequential circuit in Fig. 2 (a); we just show three time slices here. The dash-dot edges shows the second type of edges mentioned above, which couples adjacent DAGs. We have X_2 as input and X_1 as the present state node. Random variable X_6 represents the next state signal. Note that this graph is a DAG. We next prove that this TDM structure is a minimal representation, hence is a dynamic Bayesian network.

Theorem 2: The TDM structure, corresponding to the sequential circuit is a minimal I-map of the underlying switching dependency model and hence is a dynamic Bayesian network.

Proof: Let us order the random variables $\{X_{i,t_i}\}$, such that (i) for two random variables from one time t_i , X_{p,t_i} and X_{c,t_i} , where p is an input line to a gate and c is an output line to the same gate, X_{p,t_i} appears before X_{c,t_i} in this ordering and (ii) the random variables for the next time slice $t(i+1)$, $\{X_{1,t_{i+1}}, \dots, X_{n,t_{i+1}}\}$ appear after the random variables at time

slice t_i .

With respect to this ordering, the Markov boundary of a node, X_{i,t_i} , is given as follows. If X_{i,t_i}^p represents switching of an input signal line, then its Markov boundary is the variable representing the same input in time slice $X_{i,t_{i-1}}^p$. If X_{i,t_i}^s represents switching of a state signal, then its Markov boundary is the variable representing the switching at the previous time slice $X_{i,t_{i-1}}^s$. And, since the switching of any gate output line is just dependent on the inputs of that gate, the Markov boundary of a variable representing any gate output line consists of just those that represent the inputs to that gate. In the TDM structure the parents of each node are its Markov boundary elements hence the TDM is a boundary DAG. And, by Theorem 2 the TDM is a minimal I-map and thus a Bayesian network. Since nodes and the edges in the TDM defined over n time slices can be described by Eqn. 5, and Eqn. 7, the TDM is a dynamic Bayesian Network (DBN).

4.2 Conditional Probability Specifications

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

Table I. Conditional probability specification between (a) primary input line switchings at consecutive time instants: $P(x_{k,t_{i+1}}^p | x_{k,t_i}^p)$, and (b) state line switchings at consecutive time instants: $P(x_{k,t_{i+1}}^s | x_{k,t_i}^s)$

$X_{k,t_{i+1}}^p$				X_{k,t_i}^p	$X_{k,t_{i+1}}^s$				X_{k,t_i}^s
x_{00}	x_{01}	x_{10}	x_{11}		x_{00}	x_{01}	x_{10}	x_{11}	
0.5	0.5	0	0	x_{00}	1	0	0	0	x_{00}
0	0	0.5	0.5	x_{01}	0	1	0	0	x_{01}
0.5	0.5	0	0	x_{10}	0	0	1	0	x_{10}
0	0	0.5	0.5	x_{11}	0	0	0	1	x_{11}

5. INFERENCE IN TDM

In this section we present three inference schemes for Bayesian Networks. The first one is an exact inference scheme which we used to validate our model. This method is based on local message passing and is computationally expensive. This method is currently practical only for smaller circuits. For larger circuits we used the next two schemes which are hybrid schemes. Both of the schemes are based on stochastic inference. In particular, we use importance sampling based method. The first one: Evidence Pre-propagated Importance Sampling (EPIS) [Yuan and Druzdel 2003] works efficiently for prediction and also for probabilistic backtracking with evidence. The second algorithm is excellent for estimation/prediction however, degenerates for backtracking.

5.1 Exact Inference

This inference method was used to validate our model. We inferred the smallest circuit, s27, using this method. In the following discussion, the reader is requested to refer to Fig. 3 and Fig. 4 as examples, since they are drawn based on s27. The exact inference scheme is based on local message passing on a tree structure, whose nodes are subsets (cliques) of random variables in the original DAG [Cowell et al. 1999], [Hugin]. (Fig. 3(a) represents the original DAG structure of s27.) This tree of cliques is obtained from the initial

ACM Transactions on Design Automation of Electronic Systems, Vol. , No. , 20.

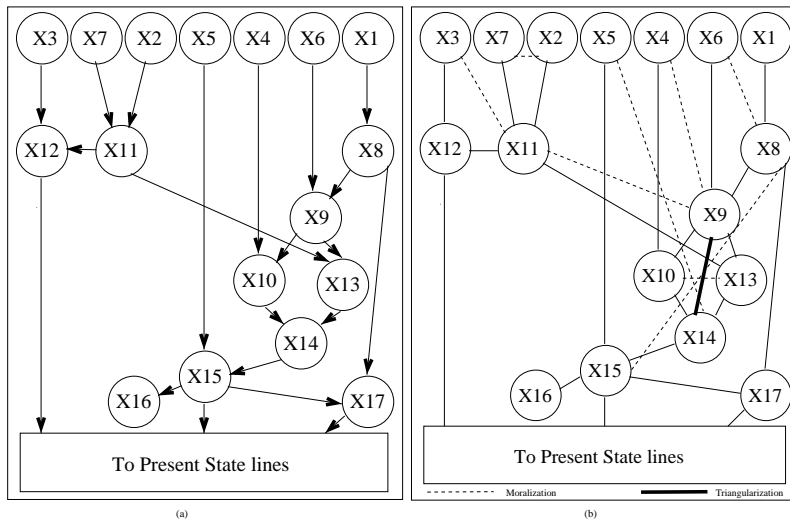


Fig. 3. (a)BN model of s27 for one time slice, (b)Triangulated undirected graph of s27 for one time slice

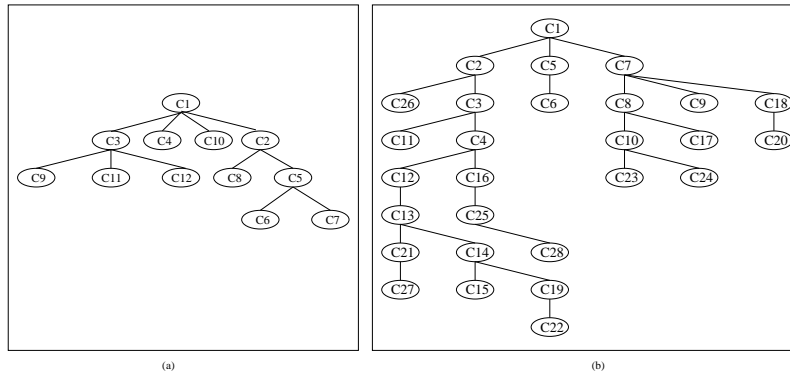


Fig. 4. Junction tree of cliques of s27 for (a)one time slice, (b)two time slices

DAG structure via a series of transformations that preserve the represented dependencies. The original DAG is first converted into an undirected Markov graph structure, which is referred to as the *moral graph*(Fig. 3(b) without the thick line), modeling the underlying joint probability distribution. This moral graph is obtained from the DAG structure, by adding undirected links between the parents of a common child node. These additional links directly capture the dependencies that were only implicitly represented in the DAG.

In a moral graph, every parent-child set form a complete subgraph. Due to the undirected nature of the moral graph, some of the *independencies* represented in the DAG would be lost, resulting in a non-minimal representation. The dependency structure is, however, preserved. This loss of minimal representation will eventually result in increased computational demands, but does not sacrifice accuracy. Next, a chordal graph is obtained from the moral graph by triangulating it (Fig. 3(b)). Triangulation is the process of breaking all cycles in the graph to be a composition of cycles over just 3 nodes by adding additional links. To control the computational demands, the goal is to form a triangulated moral graph with minimum number of additional links. Various heuristics exist for this. For instance, the Bayesian network inference software HUGIN (www.hugin.com), which we use in this work, uses efficient and accurate minimum fill-in heuristics to calculate these additional links.

Fig. 4(a) represent the junction tree of cliques for the TDM structure of s_{27} for one time slice. Cliques of the chordal graph represented in Fig. 3(b) form the nodes of the junction tree. The tree structure is useful for local message passing. Given any evidence, messages consist of the updated probabilities of the common variables between two neighboring cliques. Global consistency is automatically maintained by constructing the tree in such a way that any two cliques, sharing a set of common variables, should have these common variables present in all the cliques that lie in the connecting path between the two cliques. A junction tree with this property can be easily obtained from the same minimum fill-in heuristic algorithm that is used to triangularize the graph [Bhanja and Ranganathan 2003].

Let us consider two neighboring cliques to understand the key feature of the Bayesian updating scheme. Let two cliques A and B have probability potentials ϕ_A and ϕ_B , respectively, obtained by multiplying the conditional probabilities, in the DAG based Bayesian network, involving the nodes in each clique. Let S be the set of common nodes between cliques A and B . The two neighboring cliques have to agree on probabilities on the node

set S , which is termed 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 independent of the others.

The final junction tree of cliques for the TDM structure of s_{27} for two time slices is more complicated, as shown in Fig. 4(b). Notice the increased number of cliques. In general, the size of the maximal clique will increase. This results in increased memory requirement to store the probability potential over the nodes in the cliques; the increase is exponential in the maximal clique size. Thus, it is obvious that the exact model cannot be used for large circuits. Available memory would determine the maximum circuit size that can be modeled exactly. In this work, we use this inference only for model validation with small circuits.

5.2 Hybrid Scheme

For large circuits, a hybrid scheme, specifically the Evidence Pre-propagated Importance Sampling (EPIS) [Yuan and Druzdzel 2003], which uses local message passing and stochastic sampling, is appropriate. This method scales well with circuit size and is proven to converge to correct estimates. These classes of algorithms are also anytime-algorithms since they can be stopped at any point of time to produce estimates [Ramani and Bhanja 2004; Rejimon and Bhanja 2006]. Of course, the accuracy of estimates increases with time.

The EPIS algorithm is based on Importance Sampling that generates sample instantiations of the *whole* DAG network, i.e. all for line switching in our case. These samples are then used to form the final estimates. This sampling is done according to an importance function. In a Bayesian Network, the product of the conditional probability functions at

all nodes form the optimal importance function. Let $X = \{X_1, X_2, \dots, X_m\}$ be the set of variables in a Bayesian Network, $Pa(X_k)$ be the parents of x_k , and E^2 be the evidence set. Then, the optimal importance function is given by

$$P(X|E) = \prod_{k=1}^m P(x_k|Pa(X_k), E) \quad (9)$$

This importance function can be approximated as

$$P(X|E) = \prod_{k=1}^m \alpha(Pa(X_k)) P(x_k|Pa(X_k)) \lambda(X_k) \quad (10)$$

where $\alpha(Pa(X_k)) = P(x_k|E^+)$ and $\lambda(X_k) = P(E^-|x_k)$, with E^+ and E^- being the evidence from above and below, respectively, as defined by the directed link structure. For the proof of Eqn. 10 please refer [Yuan and Druzdzel 2003]. Calculation of λ is computationally expensive and for this, Loopy Belief Propagation (LBP) [Murphy et al. 1999] over the Markov blanket of the node is used. Yuan *et al.* [Yuan and Druzdzel 2003] proved that for a poly-tree, the local loopy belief propagation is optimal. The importance function can be further approximated by replacing small probabilities with a specific cutoff value.

5.2.1 Loopy Belief Propagation. Loopy Belief Propagation (LBP) is an approximate inference mechanism. It uses the Pearl's belief propagation algorithm [Pearl 1988] to calculate the posterior belief of each node. Here, we briefly summarize Pearl's belief propagation algorithm. Each node X computes its posterior probability based on the information obtained from its neighbors. i.e., $Bel(x) = P(X = x|E)$, where E represents the evidence set. In a polytree, any node X d-separates E into 2 subsets, E_x^+ which is the evidence connected to node X through its parent set Z and E_x^- is the evidence connected to node X through its child set Y . Now, the node X can compute its belief by separately combining the messages obtained from its parents and children.

$$Bel(x) = \alpha \lambda(x) \pi(x) \quad (11)$$

²Set of nodes that have already observed states.

where $\lambda(x)$ and $\pi(x)$ are given by

$$\lambda(x) = \prod_{U \in CH_X} \lambda_U(x) \quad (12)$$

CH_X is set of all children of X .

$$\pi(x) = \sum_{z_1, z_2, \dots, z_j} (P(x|z_1, z_2, \dots, z_j) \prod_{i=1}^j \pi_X(z_i)) \quad (13)$$

where z_1, z_2, \dots, z_j are parents of node X .

Once the node computes its belief it sends the updated messages to its neighbors. The message to the parents of node X is given by:

$$\lambda_X(w) = \sum_x \left(\sum_{z_1, z_2, \dots, z_k} (P(x|w, z_1, z_2, \dots, z_k) \prod_{i=1}^k \pi_X(z_i)) \right) \lambda(x) \quad (14)$$

The message from node X to its child is given by:

$$\pi_Y(x) = \pi(x) \prod_{C \in CH_X - Y} \lambda_C(x) \quad (15)$$

For network with loops, during each iteration all nodes calculate their outgoing messages based on the incoming messages from their neighbors during previous iteration. The iteration stops once the belief converges.

5.3 Probabilistic Logic Sampling

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

- In the circuit, which is represented as a Bayesian network, each node is selected and they are sampled.
- While inferencing, the Bayesian network the samples are grouped into sets and the observed values in each sample in a set are compared with the corresponding evidence values.

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

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

The above set of stochastic sampling strategies discussed in subsection 5.2 and 5.3 work because in a Bayesian Network the product of the conditional probability functions for all nodes is the optimal importance function. Because of this optimality, the demand on samples is low. We have found that just thousand samples are sufficient to arrive at good estimates for the ISCAS89 benchmark circuits. *Note that this sampling based probabilistic inference is non-simulative and is different from samplings that are used in circuit simulations.* In the latter, the input space is sampled, whereas in our case both the input and the line state spaces are sampled simultaneously, using a strong correlative model, as captured by the Bayesian Network. Due to this, convergence is faster and the inference strategy is stimulus-free.

5.4 Time and Space Complexity

Exact Inference: Space complexity of the exact inference of Bayesian Network is $O(n \cdot 4^{|C_{max}|})$ [Bhanja and Ranganathan 2004] where $|C_{max}|$ is the number of random variables in largest clique in the junction tree, n is the total number of random variables and usually $|C_{max}|$ is much less than n . The time complexity of the exact inference is $O(p \cdot 4^{|C_{max}|})$ where p is the number of

ACM Transactions on Design Automation of Electronic Systems, Vol. , No. , 20.

cliques in the junction tree which is also much less than n . Even though $|C_{max}|$ is less than n , for larger circuits, the exponential nature makes the inference computationally expensive.

Hybrid Inference: The space requirement of the Bayesian network representation is determined by the space required to store the conditional probability tables at each node. For a node with n_p parents, the size of the table is 4^{n_p+1} . The number of parents of a node is equal to the fan-in at the node. In the DBN model we break all fan-ins greater than 2 into a logically equivalent, hierarchical structure of 2 fan-in gates. For nodes with fan-in f , we would need $\lceil f/2 \rceil$ extra Bayesian network nodes, each requiring only 4^3 sized conditional probability table. For the worst case, let all nodes in the original circuit have the maximum fan-in, f_{max} . Then the total number of nodes in the BN structure is $n + f_{max}n/2$. Thus, the worst case space requirement is linear, i.e. $O4^3(nf_{max})$ where n is the number of nodes, f_{max} is the maximum fan-in.

The time complexity, based on the stochastic inference scheme, is also linear in n , specifically, it is $O(nN)$, where N is the number of samples, which, from our experience with tested circuits, is in the order of 1000's for circuits with 32,424 nodes in the DBN model with three time slices.

6. EXPERIMENTAL RESULTS

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

by a tool named "Hugin" [Hugin].

The tests were performed on a Pentium IV, 2.00GHz, Windows XP computer.

First, we show some results that validates the TDM model. For this, we use the *s27* benchmark circuit. Table II lists the switching estimates at each line in the circuit as computed by (i) simulation, (ii) the exact inference scheme based on tree of cliques, and (iii) the hybrid EPIS scheme. We used 10 time slices for the TDM representation. Note the excellent agreement of the exact inference scheme with simulation, thus validating that the TDM model is capturing the high order temporal and spatial correlations. The hybrid inference scheme also results in excellent estimates, close to the exact ones.

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

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

Table II. Switching probability estimates at each line of the *s27* benchmark circuit as computed by simulation, by the exact scheme, and by the hybrid EPIS method.

Nodes	Simulation	Exact (Clique Tree)	Hybrid (EPIS)
1	0.498	0.500	0.512
2	0.598	0.500	0.494
3	0.501	0.500	0.487
4	0.500	0.500	0.508
5	0.450	0.452	0.463
6	0.123	0.123	0.123
7	0.333	0.333	0.368
8	0.498	0.500	0.512
9	0.078	0.078	0.073
10	0.460	0.461	0.476
11	0.333	0.333	0.334
12	0.333	0.333	0.329
13	0.311	0.311	0.311
14	0.229	0.230	0.234
15	0.123	0.123	0.126
16	0.123	0.123	0.126
17	0.450	0.452	0.461

for larger benchmark like *s15850* ($\mu_E = 0.004$)

Note that the computation time reported for Probabilistic Logic Sampling based inference is almost ten times faster than that by TDM-EPIS based method. However, the importance of TDM-EPIS based inference scheme guarantees convergence to the exact value and also is more efficient than Logic Sampling where observation is made at any nodes other than inputs and plausible input probabilities for such an observation is estimated. This has important application in input space characterization for a desired outcome. TDM-PLS

Table III. Experimental results on switching activity estimation by Dynamic Bayesian network modeling using (a)TDM-EPIS [Yuan and Druzdzel 2003] and (b)TDM-PLS [Henrion 1988] for ISCAS'89 benchmark sequential circuits (3 time slices).

(a)					(b)				
Circuits	μ_E	\max_E	Time (s)	% of nodes $> \mu + 2\sigma$	Circuits	μ_E	\max_E	Time (s)	% of nodes $> \mu + 2\sigma$
s27	0.015	0.068	0.047	5.88	s27	0.028	0.092	0.016	5.88
s208	0.014	0.234	0.719	9.02	s208	0.014	0.224	0.281	9.02
s382	0.002	0.096	2.094	4.95	s382	0.000	0.082	0.750	7.14
s444	0.003	0.083	2.734	4.88	s444	0.005	0.067	0.843	3.90
s526	0.003	0.044	3.922	3.23	s526	0.002	0.048	1.234	1.84
s713	0.008	0.043	9.093	4.92	s713	0.009	0.067	1.968	4.70
s820	0.002	0.049	10.06	8.33	s820	0.002	0.042	2.125	4.17
s953	0.009	0.162	8.343	7.50	s953	0.012	0.185	1.922	7.95
s1196	0.001	0.050	15.06	4.81	s1196	0.001	0.043	2.735	5.17
s1238	0.001	0.038	14.62	5.00	s1238	0.003	0.035	2.703	5.00
s1423	0.010	0.127	21.12	6.15	s1423	0.012	0.114	3.266	6.02
s5378	0.002	0.402	378.68	5.08	s5378	0.001	0.389	23.128	4.98
					s15850	0.003	0.434	146.992	3.07

method is more efficient for relatively larger circuits due to its rapidity. TDM-PLS works better for circuit evaluation than for circuit characterization.

We also present estimation results with non-random inputs, in Table IV. Here two categories of biased inputs were chosen, (1) low switching inputs, where probability for switching is 0.2 (2) high switching inputs with probability of switching 0.6. The simulation results and the TDM-EPIS results were compared and errors were found to be similar to those with random inputs.

In Table V, we show the modeling for three time slices versus ten time slices for random inputs. We observe that ten time slices do not enhance the quality of estimates. This shows

Table IV. ESTIMATION WITH NON-RANDOM BIASED INPUTS using TDM-EPIS [Yuan and Druzdel

2003] (a) Low switching (b) High switching

Circuits	μ_E	\max_E	% of nodes	Circuits	μ_E	\max_E	% of nodes
			$> \mu + 2\sigma$				$> \mu + 2\sigma$
s27	0.026	0.062	0.00	s27	0.015	0.113	5.88
s208	0.009	0.173	7.38	s208	0.020	0.232	9.02
s382	0.001	0.116	7.14	s382	0.000	0.081	5.49
s444	0.007	0.091	4.39	s444	0.001	0.082	7.32
s526	0.002	0.122	5.53	s526	0.001	0.045	7.37
s713	0.012	0.102	6.26	s713	0.008	0.055	6.71
s820	0.002	0.055	4.87	s820	0.001	0.038	2.88
s953	0.009	0.157	7.27	s953	0.005	0.125	5.68
s1196	0.000	0.039	3.92	s1196	0.001	0.042	5.88
s1238	0.000	0.030	4.63	s1238	0.001	0.049	4.07
s1423	0.014	0.122	4.41	s1423	0.012	0.140	6.28
s5378	0.001	0.457	5.61	s5378	0.003	0.358	4.41

that third order temporal models are good enough for our benchmarks, which matches with the observations made in [Tsui et al. 1995; Yuan et al. 1997]. For some specific circuits 50 random input vectors will lead to wrong starting probability values. So in such circuits maximum error will keep on increasing for more time slices. We need more than 50 random vectors to get a proper starting probability values. s27 and s526 are perfect example for such circuits. In Table V, notice that \max_E is increasing for 10 time slices of s27 and s526.

Note that the switching model is extremely relevant of both static and dynamic component of power as shown in Eq. 1. In this equation, P_{dg} represents the dynamic component of power at the output of a gate g . The impact of data on dynamic component of power is encapsulated in α , the individual switching activity. The static component of power P_{sg} is

Table V. Experimental results on switching activity estimation by Dynamic Bayesian network modeling for IS-CAS'89 benchmark sequential circuits for two different time slices.

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

Table VI. Joint probabilities between specific nodes.

s1238

202				201	640				639
$0_{t-1}0_t$	$0_{t-1}1_t$	$1_{t-1}0_t$	$1_{t-1}1_t$		$0_{t-1}0_t$	$0_{t-1}1_t$	$1_{t-1}0_t$	$1_{t-1}1_t$	
0.0001	0.0008	0.0007	0.1556	$0_{t-1}0_t$	0.6649	0.0964	0.1038	0.0204	$0_{t-1}0_t$
0.0005	0.0003	0.0426	0.1965	$0_{t-1}1_t$	0.0404	0.0033	0.0072	0.0011	$0_{t-1}1_t$
0.0004	0.0449	0.0001	0.1776	$1_{t-1}0_t$	0.0462	0.0091	0.0031	0.0001	$1_{t-1}0_t$
0.0123	0.0544	0.0533	0.2599	$1_{t-1}1_t$	0.0033	0.0001	0.0002	0.0004	$1_{t-1}1_t$

dominated by $P_{leak,i}$, leakage loss in a leakage mode i . It has to be noted that each leakage mode is determined by the steady state signals that each transistor in the gate would be in. For example, in a two input (say A and B) NAND gate, the gate would have four dominant leakage mode ($i=4$: $A@_0B@_0, A@_0B@_1, A@_1B@_0$ and $A@_1B@_1$). β is the probability of each mode i and for example $\beta_1 = p(A@_0, B@_1)$ and is the joint probability of multiple signals in a gate and are dependent on the input data profile. These joint probabilities can be captured through our model as shown in Table. VI.

Hence run-time leakage power is dependent on the joint probabilities of the signals that

ACM Transactions on Design Automation of Electronic Systems, Vol. , No. , 20.

are present in a transistor stack. A combination of steady state 00 at all the signals the transistor stack produces less leakage than a steady state 11 in all of them. Also due to gate leakage the combinations 00 and 11 in two of the stacked signals also can produce considerable amount of total leakage current at sub-100 nm level technologies [Srivastava et al. 2004]. Table VI shows joint probability of a few nodes that are fan-ins to a single gate. Note that in s1238, signal 201 and 202 are fed to a transistor stack. The probability that both the signals remain at logic 0 is 0.01%, while the probability that both the signals remain at logic 1 is 26%. This result signifies that the worst case subthreshold leakage condition is highly probable. The other prominent condition for leakage (gate leakage during subthreshold) occurs when one signal (top of the stack) remains at logic 1 and the other one is at 0. In Table VI we see that the probability of signal 201 remaining at logic 0 while signal 202 remaining at logic 1 is 15.5% (considerably high). Likewise we can estimate probabilities of all the dominant leakage mode through the joint probabilities extracted through our model.

Joint probabilities are of importance in determining the upper bound of capacitive coupling noise or crosstalk. The worst case capacitive coupling noise occurs when the neighboring signals switch the opposite way. It means from $0 \rightarrow 1$ on one signal and $1 \rightarrow 0$ on the other. For example in Table VI, consider the neighboring signals 639 and 640. The probability that signal 639 switches from $0 \rightarrow 1$ while signal 640 switches from $1 \rightarrow 0$ is 0.72%, and the probability that signal 639 switches from $1 \rightarrow 0$ while signal 640 switches from $0 \rightarrow 1$ is 0.91%. Both of these conditions are favorable for placing them tightly close to each other. But the same analysis show different results in the neighboring signals 201 and 202. The probability that signal 201 switches from $0 \rightarrow 1$ while signal 202 switches from $1 \rightarrow 0$ is 4.26%, and the probability that signal 201 switches from $1 \rightarrow 0$ while signal 202 switches from $0 \rightarrow 1$ is 4.49%. Both of these conditions are considerably probable and more attention needs to be taken for placing these two neighbors. Likewise we can also

estimate the relative probabilities of logic and delay faults through the joint probabilities extracted through our model.

7. CONCLUSIONS

This paper introduces a switching activity estimation tool that encapsulates all the dependencies, in sequential circuits in reasonable time and with high accuracies. We use graphical probabilistic model (compact, minimal and dependency preserving) of the whole circuit as a single joint distribution function by a Bayesian Network. This is to be noted that the Bayesian Network based approach unifies the estimation of switching probabilities of state nodes and also the circuit nodes. We have shown results of the estimated switching activity for pseudo-random inputs as well as for biased inputs. The model handles higher order spatio-temporal dependencies between the nodes under zero-delay scenario. Our future effort will focus on studying the trade-off between various time slices, convergence issues and also the role and length of initial state probabilities on convergence and accuracy under realistic delay models. One approach that is outlined in [Manich and Figueras 1997] can be used to model switching under realistic delay.

REFERENCES

- BHANJA, S. AND RANGANATHAN, N. 2001. Dependency preserving probabilistic modeling of switching activity using bayesian networks. In *IEEE/ACM Design Automation Conference*. 209–214.
- BHANJA, S. AND RANGANATHAN, N. 2003. Switching activity estimation of vlsi circuits using bayesian networks. *IEEE Transactions on VLSI Systems* 11, 4 (Aug.), 558–567.
- BHANJA, S. AND RANGANATHAN, N. 2004. Cascaded bayesian inferencing for switching activity estimation with correlated inputs. *IEEE Transactions on VLSI Systems* 12, 12 (Dec.), 1387–1397.
- BRYANT, R. E. 1992. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys* 24, 3 (Sept.), 293–318.
- CHEN, Z. AND ROY, K. 1997. An efficient statistical method to estimate average power in sequential circuits considering input sensitivities. In *ASIC Conference and Exhibit*. 189–193.
- ACM Transactions on Design Automation of Electronic Systems, Vol. , No. , 20.

- COWELL, R. G., DAVID, A. P., LAURITZEN, S. L., AND SPIEGELHALTER, D. J. 1999. *Probabilistic Networks and Expert Systems*. Springer-Verlag New York, Inc.
- DING, C. S., TSUI, C. Y., AND PEDRAM, M. 1998. Gate-level power estimation using tagged probabilistic simulation. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems* 17, 11 (Nov.), 1099–1107.
- ERCOLANI, S., FAVALLI, M., DAMIANI, M., OLIVO, P., AND RICCO, B. 1992. Testability measures in pseudorandom testing. *IEEE Transactions on CAD* 11, 794–800.
- GENIE. Graphical network interface. URL <http://www.sis.pitt.edu/genie/genie2>.
- HACHTEL, G. D., MACII, E., PARDO, A., AND SOMENZI, F. 1994. Probabilistic analysis of large finite state machines. In *Design Automation Conference*. 270–275.
- HENRION, M. 1988. Propagation of uncertainty by probabilistic logic sampling in bayes' networks. In *Uncertainty in Artificial Intelligence*.
- HUGIN. Hugin. URL <http://www.hugin.com/>.
- KJAERULFF, U. 1995. dhugin: A computational system for dynamic time-sliced bayesian networks. *International Journal of Forecasting* 11, 89–111.
- KOZHAYA, J. N. AND NAJM, F. N. 2001. Power estimation for large sequential circuits. *IEEE Transactions on VLSI Systems* 9, 2, 400–406.
- MANICH, S. AND FIGUERAS, J. 1997. Maximizing the weighted switching activity in combinational cmos circuits under the variable delay model. In *European Design and Test Conference*. 597–602.
- MARCULESCU, D., MARCULESCU, R., AND PEDRAM, M. 2000. Theoretical bounds for switching activity analysis in finite-state machines. *IEEE Transactions on VLSI Systems* 8, 3, 335–339.
- MARCULESCU, R., MARCULESCU, D., AND PEDRAM, M. 1998. Probabilistic modeling of dependencies during switching activity analysis. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems* 17, 2 (Feb.), 73–83.
- MARCULESCU, R., MARCULESCU, D., AND PEDRAM, M. 1999. Sequence compaction for power estimation: Theory and practice. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 18, 7, 973–993.
- MURPHY, K. P., WEISS, Y., AND JORDAN, M. I. 1999. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of Uncertainty in AI*. 467–475.

- NAJM, F., GOEL, S., AND HAJJ, I. 1995. Power estimation in sequential circuits. In *32nd ACM/IEEE Design Automation Conference*. 635–680.
- NAJM, F. N. 1993. Transition density: A new measure of activity in digital circuits. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems* 12, 2 (Feb.), 310–323.
- NGUYEN, D., DAVARE, A., ORSHANSKY, M., CHINNERY, D., THOMPSON, B., AND KEUTZER, K. 2003. Minimization of dynamic and static power through joint assignment of threshold voltages and sizing optimization. In *International Symposium on Low Power Electronics and Design*. 158–163.
- PEARL, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*. Morgan Kaufmann Publishers, Inc.
- RAMANI, S. AND BHANJA, S. 2004. Anytime probabilistic switching model using bayesian networks. In *International Symposium on Low Power Electronic Design*. 86–89.
- REJIMON, T. AND BHANJA, S. 2006. A timing-aware probabilistic model for single-event-upset analysis. In *Accepted for publication in IEEE Transactions on VLSI Systems*.
- SAXENA, V., NAJM, F. N., AND HAJJ, I. N. 2002. Estimation of state line statistics in sequential circuits. *ACM Transactions on Design Automation of Electronic Systems* 7, 3, 455–473.
- SRIVASTAVA, A., SYLVESTER, D., AND BLAAUW, D. 2004. Power minimization using simultaneous gate sizing, dual-vdd and dual-vth assignment. In *Design Automation Conference*. 783–787.
- STAMOULIS, G. I. 1996. A monte-carlo approach for the accurate and efficient estimation of average transition probabilities in sequential logic circuits. In *IEEE Custom Integrated Circuits Conference*. 221–224.
- TSUI, C. Y., MONTEIRO, J., PEDRAM, M., DEVADAS, S., DESPAIN, A. M., AND LIN, B. 1995. Power estimation methods for sequential logic circuits. *IEEE Transactions on VLSI Systems* 3, 3, 404–416.
- YUAN, C. AND DRUZDZEL, M. J. 2003. An importance sampling algorithm based on evidence pre-propagation. In *Proceedings of the 19th Annual Conference on Uncertainty on Artificial Intelligence*. 624–631.
- YUAN, L. P., TENG, C. C., AND KANG, S. M. 1997. Statistical estimation of average power dissipation in sequential circuits. In *34th Design Automation Conference*.