Switching Activity Estimation of Large Circuits using Multiple Bayesian Networks*

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

Abstract

Switching activity estimation is a crucial step in estimating dynamic power consumption in CMOS circuits. In [1], we proposed a new switching probability model based on Bayesian Networks which captures accurately the various correlations in the circuit. In this work, we propose a new strategy for efficient segmentation of large circuits so that they can be mapped to Multiple Bayesian Networks (MBN). The goal here is to achieve higher accuracy while reducing the memory requirements during the computation. In order to capture the correlations among the boundaries of segments, a tree-dependent (TD) distribution is proposed between the segment boundaries such that the TD distribution is closest to the actual distribution of switching variable with some distance criterion. We use a Maximum Weight Spanning Tree (MWST) based approximation [4] using mutual information between two variables at the boundary as weight of the edge between the variables. Experimental results for ISCAS'85 circuits show that the proposed method improves accuracy significantly over other methods.

1 Introduction

Switching activity estimation strategies can be divided into two broad categories: (i) estimation by simulation and (ii) estimation by probabilistic techniques. Probabilistic techniques [9, 12, 13, 6] are fast and tractable, but typically involve assumptions about joint correlations. The 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 the mean and variance of current using probability waveforms accounting for temporal correlation with spatial independence assumptions. In a later work, Najm et al. [10] introduced the concept of transition density. However, these methods have been reported to yield inaccurate estimates when the nodes are highly correlated. An improved switching activity estimation strategy based on OBDD was proposed by Bryant [9], however, it had high space requirements. Ghosh et al. [6] modeled temporal correlation effects in a real delay model. The computational complexity was, however, extremely high. Tagged probability simulation was proposed by Ding et al. [12], which was based on local OBDD propagation with a real delay model.

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 [14] has modeled structural dependencies and Schneider *et al.* [18] used one-lag Markov model to capture temporal dependence. Tsui *et al.* [15] 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. Pair-wise correlation between circuit lines were first proposed by Ercolani et al. [11]. Marculescu et al. [20], studied temporal, spatial dependencies studying pairwise correlations. In a later pioneering work, Marculescu et al. [8], formulated higher order correlations by approximating them as a set of pair-wise correlations. We proposed LIDAG constuction [1] from a combinational circuit and proved that it corresponds to a Bayesian Network which captures higher order spatio-temporal correlations in the internal nodes. When the circuits were large we used naive segmentation strategy without really capturing the dependencies between two BNs.

In this paper, our focus is to systematically formulate the problem of using Multiple Bayesian Network for

*This work is being supported in part by a Nation Science Foundation Grant No. CCR-0098103

large circuits such that correlations between the BNs are captured effectively through approximate tree-dependent distribution at the boundaries. Results show that the Multiple BN modeling has even higher accuracy compared to naive segmentation based approach in [1] for large combinational benchmark circuits.

2 Bayesian Network Modeling

The theory behind modeling switching activity using single BN has been proposed and discussed in [1]. Some of the basic definitions and terminologies are repeated here for convenience. For more details the reader is referred to [1].

A Bayesian Network is a directed acyclic graph (DAG)and is established in the following manner. **Definition 1: [3]** Let U=$\{\alpha, \beta, \cdots\}$ be a finite set of variables taking on discrete values. Let $P(.)$ be the joint probability function over the variables in U, and let X, Y and Z be any three subsets (maybe overlapping) of U. X and Y is said to be *conditionally independent* given Z if

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

Following Pearl[3], this conditional independence amongst X, Y, and Z is denoted as $I(X, Z, Y)$ in which X and Y are said to be *conditionally independent* given Z.

The concept of conditional independence can be extended to dependencies in a graphical structure by Pearl [3] and is discussed in [1]. Moreover, given any Probability function, we can also get a set of all the I relations. This set of three tuple I's form an I-map.

If the network representation has I-map as a superset of that derived from the probability function then the network is an approximate under-representation (since the network assumes more independence relation that encoded in the probability function). If the network representation has an I-map that is subset of that encoded by the probability function then the network is an over-representation. Only when these two I-maps are equal we get an exact representation which is the minimal one that captures the independence model completely. This network representation is actually a **Bayesian Network**.

Definition 2:[1] Given a probability function P on a set of variables U, a DAG D is called a *Bayesian Network* of P if D is a minimum I-map of P.

The next step is to have DAG network representation of the switching model of a combinational circuit.

Definition 3: [1] A Logic Induced Directed Acyclic Graph (LIDAG) structure, LD, corresponding to a combinational circuit consists of nodes, X_is, representing the switching at each line and links between them is constructed as follows: The parents of a random variable representing the switching at an output line, O_i, of a gate G_i are the nodes representing switchings at the input lines of that gate. Each input line is either one of $\{I_1, \cdots, I_n\}$ or an output of another gate.

Theorem 2: The **LIDAG** structure, LD, corresponding to the combinational circuit is a minimal I-map of the underlying switching dependency model and hence is a **Bayesian Network**. **Proof:** Proof in[1].

In the LIDAG structure, each random variable represents switching activity on each line in the combinational circuit. Each variable can take four values corresponding to the four possible transitions: $\{x_{00}, x_{01}, x_{10}, x_{11}\}$. Note that this way of formulating the random variable effectively models temporal correlation since we use zero-delay model. The probability of switching at a line would be given by $P(X_i = x_{01}) + P(X_i = x_{10})$. The original Bayesian Network is converted in junction tree of cliques for computational efficiency and the conversion stages were briefly mentioned in [1]. In the next section, we want to discuss a few steps in BN computation in details which will direct us towards the need for MBN modeling.

3 BN Computation

We have already discussed the transformation of a BN which is a DAG to a junction tree of cliques in our earlier work [1]. This transformation is necessary for local message passing inference scheme. Each node of the junction tree is a clique or a collection of random variables. The same random variable may be present in more than one clique. An important property of the junction tree is that if two cliques share a random variable then all cliques in the path between these cliques contain the same random variable. This property is crucial for updating by local message passing.

Once junction tree of cliques is formed in the compilation process, we need to form the distribution function for the cliques. We then follow a two phase algorithm for message propagation. All the messages that is propagated are between neighboring cliques. Let $\{\mathbf{x_c}\}$ be the set of nodes in clique c in the junction tree. The joint probability function over these variables is denoted by $p(\mathbf{x_c})$. Let $\{\mathbf{x_s}\}$ be the set of nodes in a separator set s between two cliques in the junction tree. The joint probability function over these variables is denoted by $p(\mathbf{x_s})$. Let CS denote the set of all cliques and SS denote the set of separators in the junction tree. The joint probability function factorizes over the junction tree in the following form [2]:

$$p(x_1, \cdots, x_N) = \prod_{c \in \text{CS}} p(\mathbf{x_c}) / \prod_{s \in \text{SS}} p(\mathbf{x_s}) \quad (2)$$

A separator set s is contained in two neighboring cliques, c_1 and c_2. If we associate each of the product terms over the separators in the denominator with one its two neighboring cliques, say c_1, then we can write the joint probability function in a pure product form as follows. Let $\phi_{c_1}(\mathbf{x_{c_1}}) = p(\mathbf{x_{c_1}})/p(\mathbf{x_s})$ and $\phi_{c_2}(\mathbf{x_{c_2}}) = p(\mathbf{x_{c_2}})$, then the joint probability function as expressed as:

$$P(x_1, \cdots, x_N) = \prod_{c \in \text{CS}} \phi_c(\mathbf{x_c}) \quad (3)$$

where the factors $\phi_c(\mathbf{x_c})$ are also commonly referred to as the potential function over the nodes $\{x_c\}$ in clique c, and CS is the set of cliques. These functions, $\phi_c(\mathbf{x_c})$s, can be formed by multiplying the conditional probabilities, from the input Bayesian network specification, of nodes in the clique c.

Let us now focus on the information flow in the 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. Let S be the set of nodes that separates cliques A and B. Let us say that there is new evidence for some node. This will change the probabilities of all the other nodes such that two neighboring cliques agree on probabilities on the node set S which is their separator. To achieve this we first compute the marginal probability of S from probability potential of clique A and then use that to scale the probability potential of B as captured by Eq. 5. To achieve this we need to transmit the scaling factor along the link and this process is referred to as message passing. We have to repeat this process in the reverse direction by computing the marginal probability of S from probability potential of clique B and then use that to scale the probability potential of A. This will ensure that evidence at both the cliques are taken into account. New evidence is absorbed into the network by passing such local messages. The pattern of the message is such that the process is multi-threadable and partially parallelizable. Because the junction tree has no cycles, messages along each branch can be treated independently of the others.

$$\phi_S^* = \sum_{X \in A, X \notin S} \phi_A \quad (4)$$

$$\phi_B^* = \phi_B \frac{\phi_S^*}{\phi_S} \quad (5)$$

We use a two phase message passing scheme that can integrate all new evidence in two passes. In this scheme a clique from junction tree is selected to be the root node. In the first phase, the collection phase, all the leaf node of the tree send messages towards the root node, which are re-computed according at Eqs. 4 and 5 at each node along the way. Once messages from all the leaf nodes have been received, the second phase is initiated when messages are passed back from the root clique towards the leaves. Note that this ensures that along any one link we have messages along both directions, thus ensuring all nodes have been updated based on information from all the new evidence.

The clique size i.e. number of random variables in a clique is dependent on the number of parents that a node have and the connectivity and also on the number of added links during the triangularization. As the clique size (k) increases we suffer three problems: (i) the potential table is 4^k becomes memory intensive for each of the large cliques and (ii) formation of the joint probability function in terms of the cliques become computationally expensive(Eq.3) and (iii) the updating through Eqs. 4 and 5 becomes computationally expensive. Hence we resort to individual smaller BNs loosely coupled with each other to increase performance. In the next section, we discuss modeling based on naive coupling between individual BNs followed by Multiple BN modeling using a tree based coupling (by using TD distribution).

4 Modeling with Multiple BN

It is evident that if the circuit is considerably large or if the circuit has high connectivity between nodes, then, the size of the cliques formed becomes large. This results in large probability potential tables and might make handling large circuits in one stage impossible with limited memory resources. To handle large circuits, we first adopt a divide and conquer strategy based on naive segmentation.

4.1 Naive Segmentation

We first segment the large circuit into smaller ones and then estimate switching activity in them by ensuring consistency of probability of the nodes at the boundary. Hence, probability values on all the states of each node, which is in the boundary between two BNs are updated sequentially from one BN to the next BN. As an illustration, consider two BNs G_1 and G_2, as shown in Figure 1, which are the two smaller BNs of the initial BN 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. As described before we remove

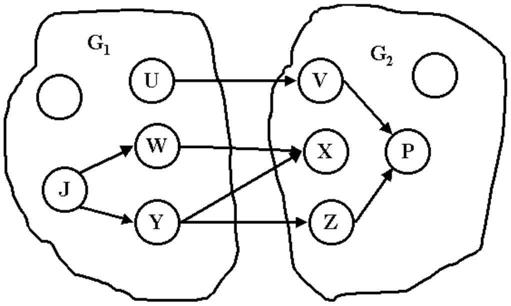

Figure 1. Segments G_1 and G_2 of a graph G with common edges.

edges (U, V), (W, X), (Y, X) and (Y, Z) from G_1, however nodes U, W and Y are retained in G_1. In G_2, we have all four edges, however, nodes U, W and X are now one of the root nodes. Moreover, we calculate probabilities for nodes U, W and Y by updating G_1 with all the input evidences as described in the previous section. We then use the same probabilities for U, W and X in G_2. It is obvious that we lose some correlations as we propagate only the singleton probabilities. One way to account for the lost correlation is to learn a BN in the boundary node and cascade it to the next BN. However, the accuracy improvement by a perfect Bayesian Network modeling may become marginal as opposed to the enormous computational time. Hence, we resort to an approximation technique which is a good compromise between accuracy and memory requirement.

4.2 MBN-TD Based Modeling

In the earlier section, we discussed the naive segmentation scheme. In the following section we propose a tree-dependent(TD) structure over the boundary nodes. This TD structure would be only an approximation of the joint probability distribution over the boundary nodes, but will ensure that most of the correlations are captured with minimal increase in memory and time requirement. Relaxation of the tree-structure constraint would entail large memory requirement. The tree-structure is an excellent compromise between memory requirement and the ability to represent correlations between the BN boundary nodes. The tree-dependent structure is formally defined next followed by proof of its optimality, which can also be found in [4].

Definition 4: 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,

$$p^t(x) = \prod_i p(x_i | x_{j(i)}) \tag{6}$$

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 segment boundary 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 Kullbuck-Leibler cross-entropy measure [5] in Eq. 7.

$$D(P, P') = \sum_x P(x) \log(P(x)/P'(x)) \tag{7}$$

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 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 [4].

Theorem 3: The projection of P on t is characterized by the equality

$$P_P^t(x_i | x_{j(i)}) = P(x_i | x_{j(i)}) \tag{8}$$

Proof: For proof see [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 4 : [4] The distance measure of Equation 7 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

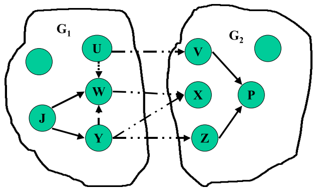

Figure 2. Consecutive BN G_1 and G_2 coupled by TD.

$$I(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log(P(x_i, x_j)/P(x_i)P(x_j)) \tag{9}$$

Proof: For proof see [3].

Hence we can use any algorithm for any MWST finding algorithm. The tree distribution thus obtained (TD) help us couple the two adjacent BNs. The steps we follow, as proposed by [4], are listed below in Figure 3.

Using a tree-structured representation ensures that storage proportional to $(r-1)r(n-1)+r-1$ [3] is used where r is the number of states (in our case r = 4) and n is the number of variables on the segment boundary which is much less than r^n which would be needed for a complete representation. Moreover, by the above algorithms, we ensure that the pairwise correlations are captured effectively and propagated to the next segments. Hence, as in Figure 2, we now construct TD distribution (by MWST) at the boundary nodes of BN G_1 and cascade this tree (marked in dotted lines) and the boundary edges (marked in dash-dot lines) to the original BN G_2.

5 Experimental Results and Conclusions

We mapped large ISCAS circuits to their corresponding LIDAG structured Bayesian Networks. The conditional probabilities are pre-determined by the type of gate connecting the parents and the child. We used HUGIN's Bayesian Network tool for compiling the junction tree and propagating the probabilities. We also performed logic simulation providing "ground truth" estimates of switching.

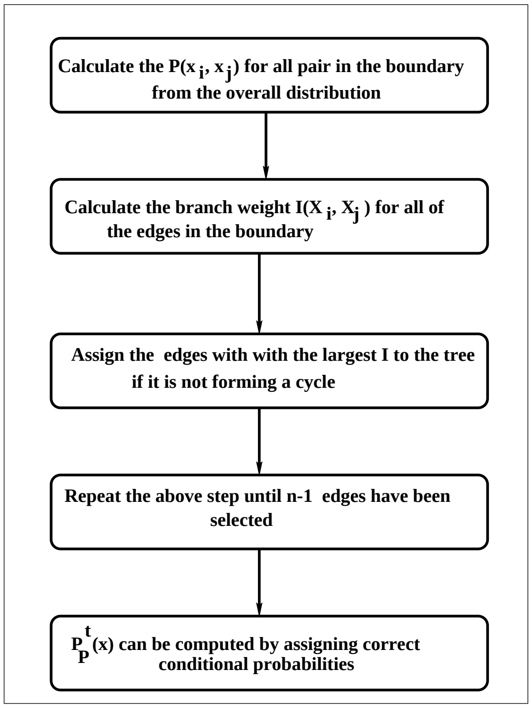
Calculate the P(x i, x j) for all pair in the boundary from the overall distribution
Calculate the branch weight I(X i, Xj) for all of the edges in the boundary
Assign the edges with with the largest I to the tree if it is not forming a cycle
Repeat the above step until n-1 edges have been selected
P_P^t(x) can be computed by assigning correct conditional probabilities

Figure 3. Steps involved in construction of MBN-TD

We presented results with all the ISCAS circuits in [1] with random inputs. Our accuracy was high and the estimation time was low. From the results, we observed that circuits which are modeled with multiple BN were less accurate than that modeled by single BN and some circuits have higher standard deviation of errors because of the error propagation through the circuits originated at the naively coupled segments. This is the motivation for the tree-dependent(TD) distribution at the BN boundaries. In this section, we compare the results that we obtained from naive segmentation with segmentation with a tree-dependent(TD) distribution between segments in Table 1. We present the mean, standard deviation and maximum error for the large circuits with TD segmentation and we compare the results with that obtained from the naive segmentation. As it can be observed that the tree-based segmentation results in increased accuracy; mean error is further reduced. Moreover, we reduce the maximum error in estimation by using tree-based segmentation over naive segmentation. As it can be observed that for all the circuits our maximum error has improved. The deviation of error has either been same or have reduced in almost all the circuits. We also achieved a large improvement in our mean error for almost all the circuits except c3540. We encountered still a high maximum error of 0.318 in the circuit c6288 which indicates that for a few sparse nodes the TD based segmentation may not be enough to capture correlation for those nodes. However, our standard deviation is quite low indicating that most of

Proceedings of the 15th International Conference on VLSI Design (VLSID'02)
0-7695-1441-3/02 $17.00 © 2002 **IEEE**

	naive segmentation			MBN-TD based segmentation		
Circuits	μ_{Err}	σ_{Err}	Max_{Err}	μ_{Err}	σ_{Err}	Max_{Err}
c432	0.006	0.031	0.290	0.002	0.032	0.197
c499	0.000	0.004	0.023	0.000	0.001	0.006
c880	0.001	0.009	0.066	0.001	0.009	0.066
c1355	0.006	0.033	0.188	0.001	0.017	0.124
c1908	0.001	0.010	0.155	0.001	0.010	0.099
c3540	0.003	0.044	0.279	0.005	0.037	0.252
c6288	0.014	0.046	0.421	0.006	0.023	0.318

Table 1. Comparison between naive and MBN-TD based segmentation.

	Schneider et al., 96*[19]	Marculescu et al., 98*[7]			MBN-TD based segmentation		
Circuit	μ_{Err}	μ_{Err}	σ_{Err}	Max	μ_{Err}	σ_{Err}	Max
c432	0.016	0.028	0.04	0.21	0.002	0.03	0.197
c499	-	0.013	0.01	0.062	0	0	0.006
c880	0.006	0.013	0.02	0.069	0.001	0.01	0.066
c1355	0.005	0.004	0	0.003	0.001	0.02	0.124
c1908	0.01	0.009	0.02	0.131	0.001	0.01	0.099
c3540	0.014	0.03	0.04	0.201	0.005	0.04	0.252
c6288	0.023	0.014	0.02	0.089	0.006	0.02	0.318

Table 2. Comparison with techniques [7, 19].

the node errors are close to the mean which is very low for c6288 circuit.

We compare the estimation by multiple Bayesian Network modeling using new TD based segmentation scheme in the boundary of the individual BNs with the BN modeling using naive coupling between individual BNs [1] in Table 1, where we see significant improvement. In Table 2, we compare our result with the estimation technique proposed by Schneider et al.[19] and estimation techniques proposed by Marculescu et al. [7]. Obviously, since these works are conducted on different machines a definite conclusion cannot be drawn. The table is provided just for qualitative analysis. Our future effort will focus on the input modeling to capture spatial correlation at the primary inputs themselves.

References

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

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

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

[4] C. K. Chow, C. N. Liu, “Approximating Discrete Probability Distributions with Dependence Trees”, *IEEE Trans. Info. Theory*, vol. 14, pp.462-467, 1968.

[5] S. Kullbuck, R. A. Leibler, “ Information and Sufficiency”, *Ann. Math. Statistics*, vol. 22, pp. 79-86.

[6] A. Ghosh, S. Devadas, K. Keutzer, and J. White, “Estimation of Average Switching Activity in Combinational and Sequential Circuits”, *Proc. 29th Design Automation Conference*, pp.253-259, June 1992.

[7] R. Marculescu, D. Marculescu, and M. Pedram, “Probabilistic Modeling of Dependencies During Switching Activity Analysis”, *revised version submitted to IEEE Trans. CAD*,

URL= `http://atrak.usc.edu/˜massoud/ sign_download.cgi?pecp-journal.ps`

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

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

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

[11] 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.

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

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

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

[15] 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, November 1993.

[16] A. Murugavel, N. Ranganathan, R. Chandramouli and S. Chavali, ”Average Power in Digital CMOS Circuits Using Least Square Estimation”, in *Proc. of Intl. Conf. on VLSI Design*, Jan 2001.

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

[18] P. Schneider, and U. Schlichtmann, “Decomposition of Boolean Functions for Low Power Based on a New Power Estimation Technique”, *Proc. 1994 Int’l Workshop on Low Power Design*, pp.123-128, April 1994.

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

[20] R. Marculescu, D. Marculescu, and M. Pedram, “Switching Activity Analysis Considering Spatiotemporal Correlations”, *Proc. ICCAD*, pp.294-299, November 1994.