Probabilistic Modeling of Dependencies During Switching Activity Analysis

Radu Marculescu, Diana Marculescu, Massoud Pedram

Department of Electrical Engineering - Systems

University of Southern California, Los Angeles, CA 90089

Abstract

Switching activity estimation in combinational circuits is addressed from a probabilistic point of view. The zero-delay model is used and under correlated input sequences, activities at the circuit outputs and internal circuit nodes are estimated using lag-one Markov Chains and conditional probabilities to manage complex spatiotemporal correlations. Two new concepts - conditional independence and isotropy of signals - are brought into attention and based on them, sufficient conditions for exact analysis of complex dependencies are given. Based on these notions, it is shown that the relative error in calculating the switching activity of a logic gate using only pairwise probabilities can be upper-bounded. In the most general case, the conditional independence problem has been shown to be ***NP****-complete and thus appropriate approximation techniques (with bounded error) are presented to estimate switching activity. Evaluations of the model and a comparative analysis of benchmark circuits demonstrates the accuracy and the practicality of the method.*

1. Introduction

With the growing need for low-power electronic circuits and systems, power analysis and low-power synthesis have become primary concerns for the CAD community. To calculate average power consumption of a gate in a synchronous CMOS circuit, one can use the well-known formula $P_{avg}(x) = 0.5 (V_{dd}^2 / T_{cycle}) C_{load}\, sw(x)$ where V_{dd} is the supply voltage, T_{cycle} is the clock cycle period, C_{load} is the load capacitance and $sw(x)$ is the switching activity of the output x of any gate in the circuit [1].

Power estimation techniques must be fast and accurate in order to be applicable in practice. Not surprisingly, these two requirements interact and at some point they become contradictory. Simulation-based techniques can provide high level of accuracy, but the run time is very high; one can extract switching activity information by exhaustive simulation on small circuits, but it is unrealistic to rely on simulation results for larger circuits. Few years ago, probabilistic techniques came into the picture and demonstrated their usefulness at least for limited purposes. The key issue was from the very beginning switching activity estimation because charging and discharging different load capacitances is by far the most important source of energy dissipation in digital CMOS circuits [2].

Common digital circuits exhibit many dependencies; the most known one is the dependency due to reconvergent fan-out among different signal lines, but even structurally independent lines may have dependencies (induced by the sequence of inputs applied to the circuit) which cannot be neglected. Accounting for all kinds of dependencies is impossible even for small circuits; consequently, for real-size circuits, only some of the dependencies have been considered and even then, only heuristics have been proposed. This is because of the difficulty in managing complex data dependencies with acceptable levels of computational work.

Besides dependencies described above (called also *spatial dependencies*), another type of correlations, namely *temporal* may appear in digital circuits. Let us consider a simple case to illustrate these issues. The circuit in Fig.1 is fed successively by three input sequences, S_1, S_2 and S_3; S_1 is an exhaustive pseudorandom sequence, S_2 is also an exhaustive sequence but generated by a 3-bit counter and S_3 is obtained by a 'faulty' 3-bit counter.

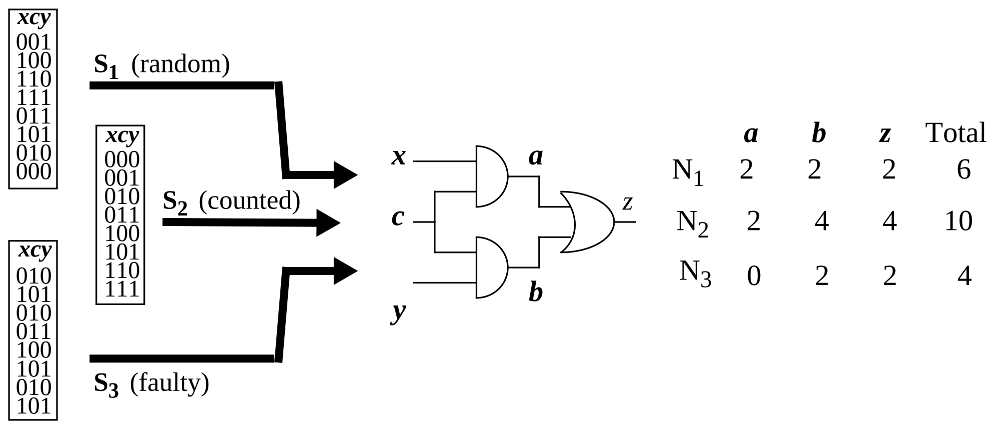


Fig.1: An illustration of spatiotemporal correlations

All three sequences have the same signal probability on lines *x*, *y* and *c* ($p = 0.5$), but even intuitively they are completely different. There are two other measures which differentiate these sequences, namely *transition* and *conditional* probabilities, and switching activity calculations should rely on them. In fact, these sequences exercise the circuit such that the number of transitions N_1, N_2, N_3 on each signal line *a*, *b*, *z* (and hence the total number of transitions) becomes quite different once we feed S_1, S_2, or S_3 respectively. In order to accurately compute the number of transitions, calculations based on signal probabilities should undoubtedly account for the influence of reconvergent fan-out, specifically in the previous figure, *a* and *b* cannot be considered independent signal lines. This problem could be solved (but only for small circuits) by building global OBDD's in terms of primary inputs, but even so, neglecting the correlations among primary inputs can lead to incorrect results. As we see in this example, assuming input independence for sequences S_2 and S_3, is an unrealistic hypothesis because the patterns in each of them are temporally correlated (for example, each pattern in sequence S_2 is obtained from the previous one by adding a binary 1). Even more than that, transitions as $0 \rightarrow 1$ or $1 \rightarrow 0$ on apparently independent signal lines (like x and c in Fig. 1) are correlated and a detailed analysis on these input streams reveals strong spatial relationship. Consequently, to compute accurately the switching activity on a node-by-node basis, one has to account for both spatial and temporal dependencies starting from the primary inputs and continuing throughout the circuit.

This paper proposes a new analytical model which accounts for spatiotemporal correlations under a zero-delay model. Its mathematical foundation is probabilistic in nature, and consists of using lag-one Markov Chains to capture different kinds of dependencies in combinational circuits [3]. Temporal correlations for the values of some signal x in two successive clock cycles are considered through a Markov Chain with only two states; spatial correlations for pairs of signals (x,y) are modeled by a four-state Markov Chain. For the first time to our knowledge, we have considered in a systematic way different kinds of dependencies in large combinational modules for both pseudorandom and highly correlated input streams. To summarize, the basic assumptions we use throughout the paper are:

- the target circuit is combinational and the logic value of any signal line x can only be 0 or 1;
- under a zero-delay model, any signal line x can switch at most once within each time step.

Under these hypotheses, this paper presents theoretical and practical evidences that *conditional independence* is a concept powerful enough to overcome difficulties arising from structural dependencies as well as highly correlated input streams [4]; more precisely, based on *conditional independence* and *signal isotropy* concepts, we give a formal proof showing that the statistics taken for pairwise correlated signals are sufficient enough to characterize larger sets of dependent signals.

The practical value of these results becomes particularly evident during optimization and synthesis for low-power; a detailed analysis presented here illustrates the importance of being accurate node-by-node (not only for the total power consumption) and identifies potential drawbacks in previous approaches when patterns feeding the inputs become highly correlated. To support the potential impact of this research, experimental results are presented for benchmark circuits.

The paper is organized as follows. First, we review the prior work relevant to our research. In Section 3 we present in detail an analytic model for switching activity estimation which accounts for spatiotemporal correlations. In Section 4 we present global and incremental propagation mechanisms for transition probabilities and transition coefficients calculation. We also provide a measure of the algorithmic complexity for the proposed propagation mechanisms. In Section 5 we improve the results given in Section 4 in the sense that we provide an enhanced incremental propagation mechanism using the concepts of conditional independence and signal isotropy. In Section 6 we give some practical considerations and report our results on common benchmark circuits. Finally, we summarize our main contribution and we indicate possible extensions of the present work.

2. Prior Work

Most of the existing work in pseudorandom testing and power estimation relies on probabilistic methods and signal probability calculations. One of the earliest works in computing the signal probabilities in combinational circuits is presented in [5]. While the algorithm is simple and general, its worse case time complexity is exponential. For tree circuits which consist of simple gates, the exact signal probabilities can be computed during a single post-order traversal of the network [6]. An algorithm, known as the cutting algorithm, which computes lower and upper bounds on the signal probability of reconvergent nodes is presented in [7]. The bounds are obtained by cutting the multiple-fanout reconvergent input lines and assigning an appropriate probability range to the cut lines and then propagating the bounds to all the other lines of the circuits by using propagation formulas for trees. The algorithm runs in polynomial time in the size of the circuits. Ercolani et al. present in [8] a procedure for propagating the signal probabilities from the circuit inputs toward the circuit outputs using only pairwise correlations between circuit lines and ignoring higher order correlations. The signal probability of a product term is estimated by breaking down the implicant into a tree of 2-input AND gates, computing the correlation coefficients of the internal nodes and hence the signal probability at the output. Similarly, the signal probability of a sum term is estimated by breaking down the implicate into a tree of 2-input OR gates.

People working in power estimation have also considered the issue of signal probability estimation. An exact procedure based on Ordered Binary-Decision Diagrams (OBDDs) [9] which is linear in the size of the corresponding function graph (the size of the graph, of course, may be exponential in the number of circuit inputs) can be found in [10]. Using an event-driven simulation-like technique, the authors describe in [10] a mechanism for propagating a set of probability waveforms throughout the circuit. Unfortunately, this approach doesn't take into account the correlations that might appear due to reconvergent fan-out among the internal nodes of the circuit. The authors in [11] use symbolic simulation to produce exact boolean conditions for switching at a particular node of the circuit. However, this approach is expensive in terms of computational cost (time and space requirements).

Recently, a few approaches which account for correlations have been proposed. Using an event-driven probabilistic simulation technique, Tsui et al. account in [12] only for first-order spatial correlations among probabilistic waveforms. Kapoor in [13] suggests an approximate technique to deal with structural dependencies, but on average the accuracy of the approach is modest. In [14] the authors rely on lag-one Markov Chains and account for temporal correlations; unfortunately, they assume independent transition probabilities among the primary inputs and use global OBDDs to evaluate switching activity (severely limiting the size of the circuits they can process).

In what follows, we introduce a new model which extends over the previous work by taking into account spatiotemporal correlations at the primary inputs of the target circuit.

3. An analytical model for dependencies

We adopt the conventional probability model which consists of the triplet (Ω, Σ, *p*), where Ω represents the sample space, Σ denotes the class of events of interest and *p* is the probability measure associated to Σ.

3.1. Temporal correlations

Let us consider first a combinational logic module fed in turn by the input vectors $V_1, V_2, ..., V_n$.

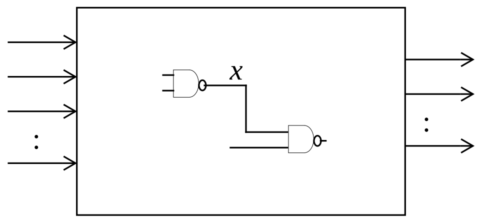


Fig.2: Temporal effects on any signal line *x*

While the input vectors $V_1, V_2, ..., V_n$ are applied at the primary inputs of the circuit at time steps 1, 2,..., *n*, the logic value of any line *x* may be 0 or 1. Hence, under a zero-delay model, *x* may switch at most once during each clock cycle.

<u>Definition 1</u>*:* If x_n is a random variable which describes the state of line *x* at any time *n*, then its behavior can be described by a lag-one Markov Chain $\{x_n\}_{n \geq 1}$ (Fig.3), over the state set Ω = {0,1}, through the transition matrix *Q* [15]:

$$Q = \begin{bmatrix} p^x_{0,0} & p^x_{1,0} \\ p^x_{0,1} & p^x_{1,1} \end{bmatrix} \quad (1)$$

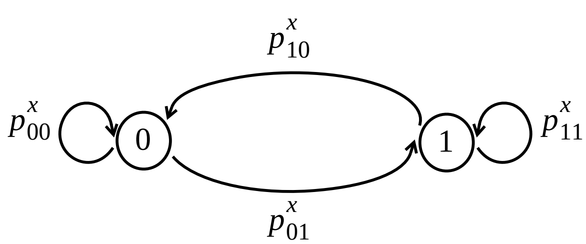


Fig.3: A lag-one Markov Chain describing temporal effects on line *x*

Every entry p^x_{ij} in the *Q* matrix represents the conditional probability of signal line *x* and may be viewed as the one-step transition probability to state *j* at step *n* from state *i* at step *n*-1. The expressions for these conditional probabilities are:

$$p^x_{i,j} = p((x_n = j)|(x_{n-1} = i)) = \frac{p((x_n = j) \cap (x_{n-1} = i))}{p(x_{n-1} = i)} \quad \forall i, j = 0, 1 \tag{2}$$

We note that Q is a *stochastic* matrix that is, every column adds to unity:

$$p^x_{0,0} + p^x_{0,1} = 1 \qquad p^x_{1,0} + p^x_{1,1} = 1 \tag{3}$$

A lag-one Markov Chain has the property that one-step transition probabilities do not depend on the 'history', i.e they are the same irrespective of the number of previous steps. The process $\{x_n\}_{n\geq 1}$ is *homogeneous* and *stationary*: indeed, because any combinational circuit is a memoryless device, having a homogeneous and stationary distribution at the primary inputs is a sufficient condition for homogeneity and stationarity to hold throughout the circuit [15]. Because the process $\{x_n\}_{n\geq 1}$ is homogeneous, then the probability distribution of the chain P may be expressed as:

$$P = (Q)^n P_0 \tag{4}$$

where P_0 is the initial distribution vector. Because the process $\{x_n\}_{n\geq 1}$ is also stationary, then relation (4) becomes:

$$P = QP \tag{5}$$

Proposition 1: The signal probabilities may be expressed in terms of conditional probabilities as follows:

$$p(x = 0) = \frac{p^x_{1,0}}{p^x_{1,0} + p^x_{0,1}} \qquad p(x = 1) = \frac{p^x_{0,1}}{p^x_{1,0} + p^x_{0,1}} \tag{6}$$

Proof: Relation (5) may be written explicitly as:

$$\begin{bmatrix} p(x=0) \\ p(x=1) \end{bmatrix} = \begin{bmatrix} p^x_{0,0} & p^x_{1,0} \\ p^x_{0,1} & p^x_{1,1} \end{bmatrix} \begin{bmatrix} p(x=0) \\ p(x=1) \end{bmatrix}$$

or $p(x = 0) = p^x_{0,0}\,p(x = 0) + p^x_{1,0}\,p(x = 1)$ and $p(x = 1) = p^x_{0,1}\,p(x = 0) + p^x_{1,1}\,p(x = 1)$, where $p(x = 1)$ represents the signal probability of line x. On the other hand we have that $p(x = 0) = 1 - p(x = 1)$, respectively $p(x = 1) = 1 - p(x = 0)$ and then relations (6) follow immediately. ■

Definition 2: We define the transition probabilities of any signal line x as follows:

$$p(x_{i \to j}) = p((x_n = j) \cap (x_{n-1} = i)) \quad \forall i, j = 0, 1 \tag{7}$$

Signal, conditional and transition probabilities associated to any signal line x are not independent measures. The following two propositions describe quantitatively the relationship between them.

Proposition 2: Transition probabilities may be expressed in terms of conditional probabilities as:

$$p(x_{0\to 0}) = \frac{p^x_{1,0}p^x_{0,0}}{p^x_{1,0}+p^x_{0,1}} \qquad p(x_{0\to 1}) = \frac{p^x_{1,0}p^x_{0,1}}{p^x_{1,0}+p^x_{0,1}} \tag{8}$$

$$p(x_{1\to 0}) = \frac{p^x_{1,0}p^x_{0,1}}{p^x_{1,0}+p^x_{0,1}} \qquad p(x_{1\to 1}) = \frac{p^x_{1,1}p^x_{0,1}}{p^x_{1,0}+p^x_{0,1}}$$

Proof: Using relation (2) and stationarity of the process, we have: $p(x_{i\to j}) = p(x=i)p^x_{i,j}$ for any values *i*, *j* = 0,1. From relation (6), the above formulas are straightforward. ■

Proposition 3: Conditional probabilities may be expressed in terms of transition probabilities as:

$$p^x_{0,0} = \frac{p(x_{0\to 0})}{p(x_{0\to 0})+p(x_{0\to 1})} \qquad p^x_{0,1} = \frac{p(x_{0\to 1})}{p(x_{0\to 0})+p(x_{0\to 1})} \tag{9}$$

$$p^x_{1,0} = \frac{p(x_{1\to 0})}{p(x_{1\to 0})+p(x_{1\to 1})} \qquad p^x_{1,1} = \frac{p(x_{1\to 1})}{p(x_{1\to 0})+p(x_{1\to 1})}$$

Proof: It suffices to use the following two identities obtained from relations (8):

$$p(x_{0\to 0})+p(x_{0\to 1}) = \frac{p^x_{1,0}}{p^x_{1,0}+p^x_{0,1}} \qquad p(x_{1\to 0})+p(x_{1\to 1}) = \frac{p^x_{0,1}}{p^x_{1,0}+p^x_{0,1}}.$$ ■

Example: Suppose that the signal line *x* takes a set of binary values described by the following string "$aababaaabb$", where $a, b \in \{0, 1\}$. To compute the signal, conditional and transition probabilities, we have to do the following calculations:

• Signal probability calculations

$p(x_n = a) = \frac{6}{10}$, $p(x_n = b) = \frac{4}{10}$ (because, out of 10 symbols, a is occurring 6 times and b is occurring 4 times).

• Conditional probability calculations

$p(x_n = a | x_{n-1} = a) = p^x_{a,a} = \frac{3}{6}$ (symbol a appears 6 times and half the time it is followed by another a).

$p(x_n = b | x_{n-1} = a) = p^x_{a,b} = \frac{3}{6}$ (symbol a appears 6 times and half the time it is followed by b).

$p(x_n = b | x_{n-1} = b) = p^x_{b,b} = \frac{1}{4}$ (symbol b appears 4 times and once it is followed by another b).

$p(x_n = a | x_{n-1} = b) = p^x_{b,a} = \frac{3}{4}$ (symbol b appears 4 times and 3 times it is followed by a).

Based on these probabilities we can define the stochastic matrix Q as: $Q = \begin{bmatrix} \frac{3}{6} & \frac{3}{6} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}$.

• Transition probability calculations

$p(x_{a \to b}) = p(x = a) \cdot p^x_{a,b} = \frac{6}{10} \cdot \frac{3}{6} = \frac{3}{10} \neq p^x_{b,a}$ (follows immediately from equations (8) and (6))

$p(x_{b \to a}) = p(x = b) \cdot p^x_{b,a} = \frac{4}{10} \cdot \frac{3}{4} = \frac{3}{10} \neq p^x_{a,b}$.

Relying on Propositions 1, 2, and 3, the relationship between signal, conditional and transition probabilities can be illustrated as in Fig.4.

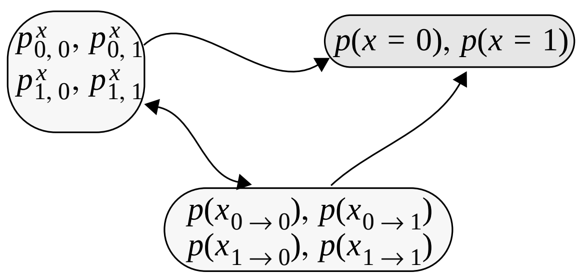


Fig.4: The paradigm of signal, conditional and transition probabilities

As we can see, to compute the signal probabilities we need less information, but the ability to derive anything else is severely limited; on the other hand, once we get either conditional or transition probabilities we have all we need for that particular signal.

Definition 3*:* For any signal line *x*, the switching activity is:

$$sw(x) = p(x_{0 \to 1}) + p(x_{1 \to 0}) = 2\frac{p^x_{1,0}\, p^x_{0,1}}{p^x_{1,0} + p^x_{0,1}} \tag{10}$$

Note: we should point out that (10) reduces to the well-known formula $sw(x) = 2 \cdot p(x = 1) \cdot [1 - p(x = 1)]$ *only if* the events are *temporally uncorrelated*. As long as we deal with temporally correlated signals, the exact relationship (10) should be used.

3.2. Spatial correlations

This type of correlations has two important sources:

• *Structural dependencies* due to reconvergent fanout in the circuit;

• *Input dependencies* that is, spatial and/or temporal correlations among the input signals which are the result of the actual input sequence applied to the target circuit.

Referring to the combinational module in Fig.5, lines *x* and *y* are obviously correlated due to the reconvergent fanout; on the other hand, even independent signal lines like the primary inputs of this module may also become correlated due to a particular input sequence (as is the case with sequences S_2 and S_3 in Fig.1when structurally independent lines *x* and *c* become correlated).

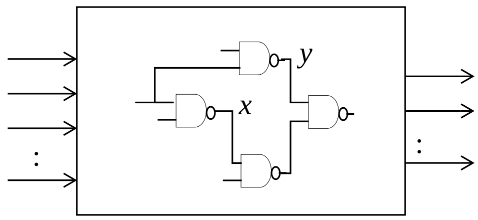


Fig.5: Two spatially correlated signal lines *x* and *y*

To take into account the exact correlations is practically impossible even for small circuits. To make this problem more tractable, we allowed only *pairwise correlated signals*, which is undoubtedly an approximation, but provides good results in practice. Consequently, we considered the correlations for all 16 possible transitions of a pair of signals (*x*,*y*) and modeled it as a lag-one Markov Chain with 4 states (denoted by 0, 1, 2, 3 which stand for encoding 00, 01, 10, 11 of (*x*,*y*)).

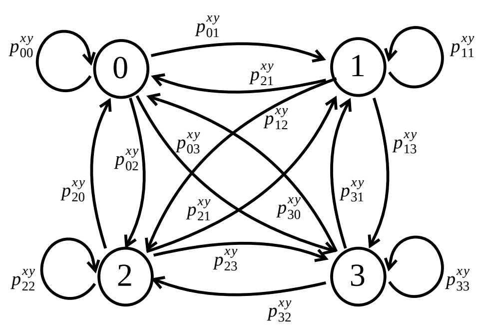


Fig.6: A lag-one Markov Chain describing spatial correlations between lines *x* and *y*

Definition 4*:* We define the conditional probability which corresponds to the pair of signals (*x*,*y*) (denoted by $p^{xy}_{a,b}$) as:

$$p^{xy}_{a,b} = p(x_n = k \cap y_n = l | (x_{n-1} = i \cap y_{n-1} = j)) \tag{11}$$

where *a, b* = 0, 1, 2, 3, *a* being encoded as *ij* and *b* as *kl*. It basically describes the probability that the pair of signals (*x, y*) goes from state *ij* at time *n*-1 to state *kl* at time step *n*.

Ercolani et al. consider in [8] structural dependencies between any two signals in a circuit, through the signal correlation coefficients (*SCs*); these coefficients can be expressed as:

$$SC^{xy}_{ij} = \frac{p(x = i \cap y = j)}{p(x = i)p(y = j)} \tag{12}$$

where *i*, *j* = 0,1. Assuming that higher order correlations of two signals to a third one can be neglected, they use the following approximation:

$$p(x = i \cap y = j \cap z = k) = \frac{p(x = i \cap y = j)p(x = i \cap z = k)p(y = j \cap z = k)}{p(x = i)p(y = j)p(z = k)}$$

Differently stated, the correlation coefficient among three signals was defined as:

$$SC_{ijk}^{xyz} = \frac{p(x = i \cap y = j \cap z = k)}{p(x = i)p(y = j)p(z = k)} \quad (13)$$

which is then equal to:

$$SC_{ijk}^{xyz} = SC_{ij}^{xy}SC_{ik}^{xz}SC_{jk}^{yz} \quad (14)$$

Our approach is more general; in order to capture the spatial correlations between signals, for each pair of signals (x,y) and for all possible transitions, we consider the transition correlation coefficients (*TCs*).

Definition 5*:* We define the *TC* for two signals x, y as:

$$TC_{ij,\,kl}^{xy} = \frac{p(x_{n-1} = i \cap x_n = k \cap y_{n-1} = j \cap y_n = l)}{p(x_{n-1} = i \cap x_n = k)p(y_{n-1} = j \cap y_n = l)} \quad (15)$$

where $i, j, k, l = 0, 1$.

Note: considering two spatially correlated signals a and b, based on *TCs* defined above we have for instance

$p(a_{0\to1}b_{1\to0}) = p(a_{0\to1}) \cdot p(b_{1\to0}) \cdot TC_{01,\,10}^{ab}$ instead of

$p(a_{0\to1}b_{1\to0}) = p(a_{0\to1}) \cdot p(b_{1\to0})$ as it would be the case if a and b were uncorrelated.

Proposition 4*:* For every pair of signals (x,y) and all possible values $i, j = 0, 1$, the following holds:

$$SC_{ij}^{xy} = \sum_{k,\,l = 0,\,1} TC_{ij,\,kl}^{xy} \frac{p(x_{i\to k})p(y_{j\to l})}{p(x = i)p(y = j)} \quad (16)$$

Proof: For the four-state Markov Chain in Fig.6 and relation (11) we have that $\sum_{b=0}^{3} p_{a,\,b} = 1$ for every value of a; that means

$$\sum_{k,\,l = 0,\,1} p(x_n = k \cap y_n = l | x_{n-1} = i \cap y_{n-1} = j) = 1$$

But, according to the definition of conditional probabilities

$$p(x_n = k \cap y_n = l | x_{n-1} = i \cap y_{n-1} = j) = \frac{p(x_{i\to k} \cap y_{j\to l})}{p(x_{n-1} = i \cap y_{n-1} = j)}$$

and then

$$\sum_{k,\,l = 0,\,1} \frac{p(x_{i\to k} \cap y_{j\to l})}{p(x_{n-1} = i \cap y_{n-1} = j)} = 1$$

Hence, from the above relation, applying (12) and (13) we get

$$\sum_{k,\, l\, =\, 0,\, 1} \frac{p(x_{i \to k})p(y_{j \to l})TC^{xy}_{ij,\, kl}}{p(x = i)p(y = j)SC^{xy}_{ij}} = 1$$

and hence the required relation is satisfied.■

<u>Proposition 5</u>: For every pair of signals (*x*,*y*) the following equations hold:

$$\sum_{j\, =\, 0,\, 1} SC^{xy}_{ij} p(y = j) = 1 \qquad \forall\, i = 0, 1 \tag{17}$$

$$\sum_{i\, =\, 0,\, 1} SC^{xy}_{ij} p(x = i) = 1 \qquad \forall\, j = 0, 1.$$

The set of 4 equations and 4 unknowns SC^{xy}_{ij} $i, j = 0, 1$ is indeterminate. Moreover, the matrix of the system has the rank ≤ 3 in non-trivial cases (i.e. when none of the signal probabilities is 1).

Proof: From the definition of *SC*, we get:

$$\sum_{j\, =\, 0,\, 1} SC^{xy}_{ij} p(y = j) = \frac{1}{p(x = i)} p((x = i) \sum_{j\, =\, 0,\, 1} (y = j)) = 1 .$$

The second equation follows in a similar manner.■

<u>Proposition 6</u>: For every pair of signals (*x*,*y*) the following equations hold:

$$\sum_{j,\, l\, =\, 0,\, 1} TC^{xy}_{ij,\, kl} p(y_{j \to l}) = 1 \qquad \forall\, i, k = 0, 1; \tag{18}$$

$$\sum_{i,\, k\, =\, 0,\, 1} TC^{xy}_{ij,\, kl} p(x_{i \to k}) = 1 \qquad \forall\, j, l = 0, 1.$$

The above set of 8 equations and 16 unknowns $TC^{xy}_{ij,kl}$ $i, j, k, l = 0, 1$ is indeterminate; the matrix of the system has the rank ≤ 7 in non-trivial cases (i.e. none of the transition probabilities is 1).

Proof: Similar to the proof for Proposition 5, but using the definition of *TC*.■

The last two propositions are very important from a practical point of view. The set of equations involving *SCs* may be solved knowing only SC^{xy}_{11} for example, and that was the approach taken by Ercolani et al. in [8] (although, no similar analysis appeared in their original paper). In the more complex case involving *TCs*, we need to know at least 9 out of 16 coefficients in order to deduce all other values.

4. Propagation Mechanisms

In what follows we ignore higher order correlations that is, correlations between any number of signals are expressed only in terms of pairwise correlation coefficients; the same assumption was used in [8], but only for signal correlation coefficients.

Definition 6: We define the *TC* among three signals as:

$$TC^{xyz}_{ijk,\,lmn} = \frac{p(x_{i\to l}y_{j\to m}z_{k\to n})}{p(x_{i\to l})p(y_{j\to m})p(z_{k\to n})} \tag{19}$$

Neglecting higher order correlations, we therefore assume that the following holds for any signals *x, y, z* and any values *i, j, k, l, m, n* = 0, 1:

$$TC^{xyz}_{ijk,\,lmn} = TC^{xy}_{ij,\,lm}TC^{yz}_{jk,\,mn}TC^{xz}_{ik,\,ln} \tag{20}$$

Definition 6 and relation (20) may be easily extended to any number of signals. Based on the above assumption, we use an OBDD-based procedure for computing the transition probabilities and for propagating the *TCs* through the network. The main reason for using the OBDD representation for a signal is that it is a canonical representation of a Boolean function and that it offers a disjoint cover which is essential for our purposes. Depending on the set of signals with respect to which we represent a node of the boolean network, two approaches may be used:

- The *global approach*: for each node, we build the OBDD in terms of the primary inputs of the circuit;
- The *incremental approach*: for each node, we build the OBDD in terms of its immediate fanin and propagate the transition probabilities and the *TCs* through the boolean network.

The first approach is more accurate, but requires much more memory and run time; indeed, for many large circuits, it is nearly impractical. The second one offers accurate enough results whilst being more efficient as far as memory requirements and run time are concerned.

4.1. Computation of transition probabilities

Let *f* be a node in the boolean network represented in terms of *n* (immediate fanin or primary input) variables $x_1, x_2, ..., x_n$; *f* may be defined through the following two sets of OBDD paths:

- Π_1 - the set of all OBDD paths in the ON-set of *f*;
- Π_0 - the set of all OBDD paths in the OFF-set of *f*;

Some of the approaches reported in the literature (e.g. [11]), use the XOR-OBDD of *f* at two consecutive time steps to compute the transition probabilities. We consider instead only the OBDD of *f* and through a dynamic programming approach, we compute the transition probabilities more efficiently.

Based on the above representation, the event '*f* switching from value *i* to value *j*' (*i, j* = 0, 1), may be written as:

$$f_{i\to j} = \bigcup_{\pi\in\Pi_i}\bigcup_{\pi'\in\Pi_j}\bigcap_{k=1}^{n} x_{k_{i_k\to j_k}} \tag{21}$$

where i_k, j_k are the values of variable x_k on the paths π and π' respectively (that is $x_k = i_k$ for path π, $x_k = j_k$ for path π', where $i_k, j_k = 0, 1, 2$, where 2 stands for don't care values) for each $k = 1, 2,..., n$. In other words, this event basically represents the *union* over all possible switchings from a path ($i_1, i_2,..., i_n$) to a path ($j_1, j_2,..., j_n$). Thus, the probability that f switches from i to j may be expressed as:

$$p(f_{i \to j}) = p\left(\bigcup_{\pi \in \Pi_i} \bigcup_{\pi' \in \Pi_j} \bigcap_{k=1}^{n} x_{k_{i_k \to j_k}}\right) \tag{22}$$

Applying the property of disjoint events (which is satisfied by the collection of paths in the OBDD), the above formula becomes:

$$p(f_{i \to j}) = \sum_{\pi \in \Pi_i} \sum_{\pi' \in \Pi_j} p(\bigcap_{k=1}^{n} x_{k_{i_k \to j_k}}) \tag{23}$$

However, since the variables x_k may *not* be spatially independent of one another, the probability of a path to 'switch' from ($i_1, i_2,..., i_n$) to ($j_1, j_2,..., j_n$) *cannot* be expressed as the product of transition probabilities for individual variables. Instead, we will use the following result which holds if we neglect higher order correlations.

Proposition 7: If relation (20) is true for any three signals in the set $\{x_1, x_2,..., x_n\}$, then:

$$p(\bigcap_{k=1}^{n} x_{k_{i_k \to j_k}}) = \prod_{k=1}^{n}\left(p(x_{k_{i_k \to j_k}}) \prod_{1 \le k < l \le n} TC^{x_k x_l}_{i_k i_l, j_k j_l}\right) \tag{24}$$

Proof: Follows directly from relation (20) by induction on the number of variables.■

According to this result, the transition probability of the signal f for any values $i, j = 0, 1$ satisfies the following:

Proposition 8: The transition probability of a signal f from state i to state j ($i, j = 0, 1$) is:

$$p(f_{i \to j}) = \sum_{\pi \in \Pi_i} \sum_{\pi' \in \Pi_j} \prod_{k=1}^{n}\left(p(x_{k_{i_k \to j_k}}) \prod_{1 \le k < l \le n} TC^{x_k x_l}_{i_k i_l, j_k j_l}\right) \tag{25}$$

Proof: Follows immediately by applying Proposition 7 to formula (23).■

Though this expression seems to be very complicated, its complexity is within reasonable bounds. We will show that it is not necessary to enumerate all *pairs* of paths in the OBDD (which would provide a quadratic complexity in the number of paths in the OBDD), but for a fixed path in Π_i the computation may be done in linear time in terms of the OBDD-nodes.

For the incremental approach, we need a mechanism not only for computing the transition probabilities (that is probabilities $p(x_{k_{i_k \to j_k}})$ in (25)), but also for propagating the *TCs* (coefficients $TC^{x_k x_l}_{i_k i_l, j_k j_l}$ in (25)) through the boolean network. For a given node in the circuit, it is only necessary to propagate the *TC*s of the output with respect to the signals on which the inputs depend. The dependency between an input and another signal may have as a cause either a reconvergent fanout or a propagated primary input dependency.

4.2. Propagation of transition correlation coefficients

Let f be a node with immediate inputs $x_1, x_2, \ldots, x_n$ and x a signal on which at least one of the inputs $x_1, x_2, \ldots, x_n$ depends. According to the definition of the *TCs*, for every $i, j, p, q = 0, 1$ possible values of f and x respectively, we have:

$$TC^{fx}_{ip,\, jq} = \frac{p(f_{i \to j} x_{p \to q})}{p(f_{i \to j})\, p(x_{p \to q})} \tag{26}$$

Since the transition probabilities for f and x are already computed at this point, the only problem is to compute the probability of both f and x switching from i to j and from p to q respectively. We get the following important result:

Proposition 9: The *TC* between signals f and x, for any values $i, j, p, q = 0, 1$ may be expressed as:

$$TC^{fx}_{ip,\, jq} = \frac{\sum_{\pi' \in \Pi_j} \sum_{\pi' \in \Pi_j} \prod_{k=1}^{n} \left(TC^{x_k x}_{i_k p,\, j_k q}\, p(x_{k_{i_k \to j_k}}) \prod_{1 \le k < l \le n} TC^{x_k x_l}_{i_k i_l,\, j_k j_l} \right)}{p(f_{i \to j})} \tag{27}$$

Proof*:* Using the representation of the event 'f switches from i to j' given in (21), we obtain the following for the event 'f switches from i to j and x switches from p to q *simultaneously*':

$$f_{i \to j} x_{p \to q} = \left(\bigcup_{\pi \in \Pi_i} \bigcup_{\pi' \in \Pi_j} \bigcap_{k=1}^{n} x_{k_{i_k \to j_k}} \right) x_{p \to q}$$

and:

$$p(f_{i \to j} x_{p \to q}) = p(\bigcup_{\pi' \in \Pi_j} \bigcup_{\pi' \in \Pi_j} \left\{ x_{p \to q} \bigcap_{k=1}^{n} x_{k_{i_k \to j_k}} \right\})$$

Applying the disjointness property of the paths, we get:

$$p(f_{i \to j} x_{p \to q}) = \sum_{\pi \in \Pi_i} \sum_{\pi' \in \Pi_j} p(x_{p \to q} \bigcap_{k=1}^{n} x_{k_{i_k \to j_k}})$$

Since the variables x_i may not be independent and, furthermore, some of them may depend on x, we need to apply the result provided by Proposition 7 for the set of n+1 variables $\{x_1, x_2, \ldots, x_n, x\}$:

$$p(f_{i \to j} x_{p \to q}) = \sum_{\pi \in \Pi_i} \sum_{\pi' \in \Pi_j} p(x_{p \to q}) \prod_{k=1}^{n} \left(TC^{x_k x}_{i_k p, j_k q} p(x_{k_{i_k \to j_k}}) \prod_{1 \le k < l \le n} TC^{x_k x_l}_{i_k i_l, j_k j_l} \right)$$

Thus, the *TC* between *f* and *x* in (27) follows immediately.■

4.3. Complexity issues

In order to assess the complexity claimed in Section 4.1, let us define the following notation:

$$f_{\pi \to j} = \bigcup_{\pi' \in \Pi_j} \bigcap_{k=1}^{n} x_{k_{i_k \to j_k}} \tag{28}$$

such that $f_{i \to j} = \bigcup_{\pi \in \Pi_i} f_{\pi \to j}$ (i, j = 0, 1 and i_k, j_k are the values of variable x_k on paths π, π' respectively). Using the disjointness property of the paths in the OBDD, the corresponding probability is:

$$p(f_{\pi \to j}) = \sum_{\pi' \in \Pi_j} p\left(\bigcap_{k=1}^{n} x_{k_{i_k \to j_k}}\right)$$

Since the path π is fixed, the above probability may be computed using the OBDD in the same way as a signal probability. The idea is that, using Shannon decomposition, the signal probability (and hence the above probability) may be computed in linear time in the number of the OBDD nodes. Thus, $f_{\pi \to j}$ may be decomposed as follows:

$$f_{\pi \to j} = x_{k_{i_k \to 0}} f^{\bar{x}_k}_{\pi \to j} + x_{k_{i_k \to 1}} f^{x_k}_{\pi \to j} \tag{29}$$

where $f^{\bar{x}_k}_{\pi \to j}$, $f^{x_k}_{\pi \to j}$ are the cofactors with respect to $\bar{x}_k$ and x_k, respectively. Based on this recursive decomposition, we may also write a similar relation for the corresponding probabilities, taking also into account the possible existing correlations:

$$p(f_{\pi \to j}) = p(x_{k_{i_k \to 0}}) p(f^{\bar{x}_k}_{\pi \to j}) \prod_{k < l \le n} TC^{x_k x_l}_{i_k i_l, 0 j_l} + p(x_{k_{i_k \to 1}}) p(f^{x_k}_{\pi \to j}) \prod_{k < l \le n} TC^{x_k x_l}_{i_k i_l, 1 j_l} \tag{30}$$

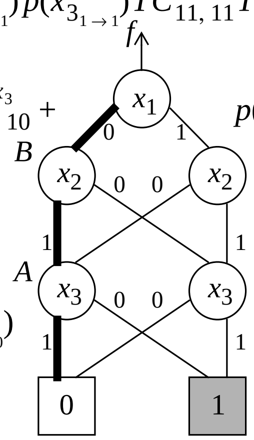
Having computed this probability for each path π, we get immediately the corresponding transition probabilities and hence the switching activity. Thus, for a fixed path π, the complexity is $O(n^2 N)$ where n is the number of variables and N is the number of nodes in the OBDD. The n^2 factor comes from the necessity of taking into account the correlations: besides the transition probabilities, we also have to keep track of the *TCs* involved on each path. There is a number of $\binom{n}{2}$ factors in the product, thus the complexity is quadratic in the number of variables.

Hence, overall, for all the paths in Π_i, the time complexity is $O(n^2NP)$ where P is the number of paths in the OBDD. In the incremental approach, this is within reasonable limits since usually n does not exceed 3 or 4 variables in the immediate fanin of the node.

Example: Let's consider the following function: $f = x_1 \oplus x_2 \oplus x_3$ and its OBDD representation from Fig.7. Suppose $i = 0$, $j = 1$ and $\pi = (0\ 1\ 1)$ is a fixed path in the OFF-set Π_0 of f. We can compute the probability given in (30) (that is, the probability of the event '$f = x_1 \oplus x_2 \oplus x_3$ switches to value 1 from the path $\pi = (0\ 1\ 1)$ in the OFF-set Π_0') by using a bottom-up parsing of the OBDD from the leaf labelled 1 to the root. We adopt a dynamic programming approach in which at each level we use the results computed at lower levels. For each node, the partial results are shown in Fig.7. For instance, in the case of node A (which corresponds to cofactoring f with respect to $\bar{x}_1 x_2$ or $x_1 \bar{x}_2$) variable x_3 must change from 1 to 0, and therefore node A is labeled with $p(x_{3_{1 \to 0}})$. At node B (corresponding to cofactoring f with respect to $\bar{x}_1$), we have two alternatives: either x_2 switches from 1 to 1 and x_3 from 1 to 0, or x_2 switches from 1 to 0 and x_3 from 1 to 1. Because these transitions are not independent, for each alternative we have to use the corresponding *TC*s as shown in Fig.7. The same operations are performed for any other path in Π_0, thus allowing us to compute in the same manner all the transition probabilities and hence the switching activity (with relation (10)). A similar approach can be further used to propagate the *TCs* between f and any other signal x.

$$\begin{aligned} p(f_{\pi \to j}) = {} & p(x_{1_{0\to 0}})p(x_{2_{1\to 1}})p(x_{3_{1\to 0}})TC^{x_2x_3}_{11,\,10}TC^{x_1x_3}_{01,\,00}TC^{x_1x_2}_{01,\,01} + \\ & p(x_{1_{0\to 0}})p(x_{2_{1\to 0}})p(x_{3_{1\to 1}})TC^{x_2x_3}_{11,\,01}TC^{x_1x_3}_{01,\,01}TC^{x_1x_2}_{01,\,00} + \\ & p(x_{1_{0\to 1}})p(x_{2_{1\to 0}})p(x_{3_{1\to 0}})TC^{x_2x_3}_{11,\,00}TC^{x_1x_3}_{01,\,10}TC^{x_1x_2}_{01,\,10} + \\ & p(x_{1_{0\to 1}})p(x_{2_{1\to 1}})p(x_{3_{1\to 1}})TC^{x_2x_3}_{11,\,11}TC^{x_1x_3}_{01,\,11}TC^{x_1x_2}_{01,\,11} + \end{aligned}$$

$$p(f_{\pi \xrightarrow{\bar{x}_1} j}) = p(x_{2_{1\to 1}})p(x_{3_{1\to 0}})TC^{x_2x_3}_{11,\,10} + p(x_{2_{1\to 0}})p(x_{3_{1\to 1}})TC^{x_2x_3}_{11,\,01}$$



$$p(f_{\pi \xrightarrow{x_1} j}) = p(x_{2_{1\to 0}})p(x_{3_{1\to 0}})TC^{x_2x_3}_{11,\,00} + p(x_{2_{1\to 1}})p(x_{3_{1\to 1}})TC^{x_2x_3}_{11,\,11}$$

$$p(f_{\pi \xrightarrow{x_1\bar{x}_2} j}) = p(f_{\pi \xrightarrow{\bar{x}_1x_2} j}) = p(x_{3_{1\to 0}})$$

$$p(f_{\pi \xrightarrow{x_1x_2} j}) = p(f_{\pi \xrightarrow{\bar{x}_1\bar{x}_2} j}) = p(x_{3_{1\to 1}})$$

Fig.7: Probability calculations for $f = x_1 \oplus x_2 \oplus x_3$

5. An Axiomatic Approach to Conditional Probability

In this section we discuss some practical limitations regarding the mechanism described in Section 4. In particular, we introduce two new concepts, that is conditional independence and signal isotropy, which allow us to overcome these practical limitations.

5.1. Issues in Performance Management

In real examples, we may have to estimate power consumption in large circuits like ISCAS benchmarks C6288, C7552, 32-bit multipliers, etc. where global approaches are totally impractical; in such cases, incremental approaches based on correlation coefficients are still applicable, despite the significant amount of CPU time they need for switching activity analysis [3]. Surprisingly enough, there are other circuits, much simpler as gate count and internal structure, which raise a lot of problems in terms of run time. In these cases, the incremental approaches need a large number of backtracks in order to compute the correlations among different signals and in some sense they "degenerate" to global approaches, that is, they tend to behave almost alike at least as far as the run time is concerned.

To begin with, let us consider ordinary tree circuits with k primary inputs consisting of common type gates (two inputs ANDs, ORs, XORs, etc.). At each level j ($1 < j \leq \log_2(k)$) we need to compute for each gate $(4^j - 1) / 3$ correlation coefficients, which adds up to a total of $\theta(k^2)$ calculations for the entire circuit. The running time for tree circuits is thus about 4-5 times that of non-tree circuits with the same number of gates and circuit inputs. This worst-case computation requirement is not present in non-tree circuits.

The degree in which the signals are correlated is reflected in the actual values of correlation coefficients; for instance, given $TC^{xy}_{ij,\,kl} = 1$, $TC^{zt}_{ij,\,kl} = 4$ and $TC^{uv}_{ij,\,kl} = 256$, then we may say that the pairs (x, y), (z, t) and (u, v) are uncorrelated, slightly correlated and highly correlated, respectively. In general, large values for the coefficients cause a lot of problems in the propagation mechanisms of the coefficients, the main rationale behind this being the approximation formulae used throughout the calculations. Accurate estimation of the switching activity is particularly important in low-power design scenarios when we are interested mostly in node-by-node comparisons among different nodes in the boolean network rather than the total power consumption in the circuit; this need precludes the classical approaches (which do not account for correlations) to have any success in real applications and made us aware of the importance of high signal correlations.

Highly correlated signals may arise everywhere in the circuits, even starting at the primary inputs; for example, suppose we want to compute the dot-product between two vectors x and y using the following piece of code (*x* and *y* are assumed to contain 10 random components between 0 and 1000):

```
for(i=0; i<10; i++) z=z+x[i]*y[i];
```

This instruction has been translated into assembly code as follows:

```
CPU Pentium
#TEST1#24: z=z+x[i]*y[i];
cs: 006A 8BDE   mov bx, si
cs: 006C 03DB   add bx, bx
cs: 006E 8D46EA lea ax, [bp-16]
cs: 0071 03D8   add bx, ax
:
:
cs: 0086 46     inc si
cs: 0087 83FE0A comp si, 000A
cs: 008A 7CDE   jl #TEST1#24 (006A)
```

We went deeper into details and we monitored the actual values at the primary inputs of a virtual 16-bit adder (Fig.8(a)) which would perform the **add** operation (e.g. instructions at addresses 006C, 0071 and so on). We give in Fig.8(b) the values at the primary inputs of this adder only for the first iteration in this loop:

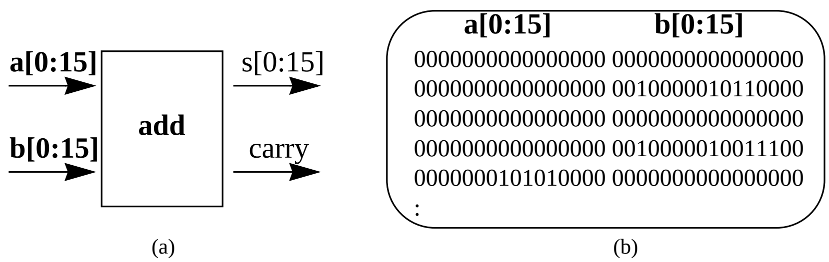


Fig.8: An example involving highly correlated signals

Analyzing the whole sequence coming from all 10 iterations we found that it is a highly correlated one. Assuming input independence is therefore incorrect and we give in Fig.9 the overestimation of switching activity per node one would make by ignoring the actual input statistics.

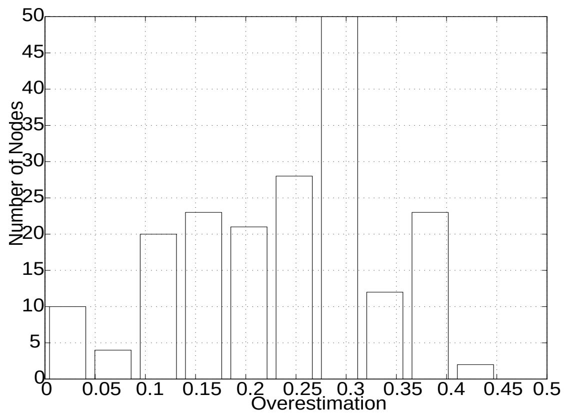


Fig.9: Switching activity overestimation by ignoring input statistics

As we can see, more than 70% of the nodes are significantly overestimated. This is a typical case which may arise in practice; therefore, in order to "make the common case accurate," we need a really good mechanism to control the error level throughout the circuit. Unfortunately, the spatiotemporal hypothesis alone does not provide a bounding value for the error.

These two limitations, namely accuracy degradation for highly correlated signals and excessive running time for tree circuits, stimulated us to further investigate stronger concepts able to overcome these drawbacks.

5.2. Conditional Independence and Signal Isotropy

Definition 7: (Conditional Independence)

Let (Ω, Σ, p) be a discrete probability space and let A, B and C be three events; the events A and B are *conditionally independent* with respect to C iff

$$p(A \cap B|C) = p(A|C) \cdot p(B|C) \tag{31}$$

The above definition may be extended to any number of digital signals as follows:

Definition 8: Given the set of n signals $\{x_1, x_2,..., x_n\}$ and an index i ($1 \le i \le$ n), we say that the subset $\{x_1, x_2,..., x_{i-1}, x_{i+1},..., x_n\}$ is *conditionally independent* with respect to x_i if the following holds:

$$p(x_1 \cap x_2 \ldots \cap x_{i-1} \cap x_{i+1} \ldots \cap x_n | x_i) = \prod_{1 \le j \le n,\ j \ne i} p(x_j | x_i) \tag{32}$$

Note: It should be pointed out that if the set $\{x_1, x_2,..., x_{i-1}, x_{i+1},..., x_n\}$is conditionally independent with respect to x_i, it might not be conditionally independent with respect to $\bar{x}_i$. However, the corresponding set in which *any* variable (or subset of variables) is complemented, is still conditionally independent with respect to x_i if the conditions from Definition 8 are met.

Using the notion of support of a boolean function (i.e. the set of variables on which the function depends), we give the following definition:

Definition 9: (Logic Independence)

Two boolean functions f and g are said to be *logically independent* (notation $f \perp g$) iff $Sup(f) \cap Sup(g) = \varnothing$; if they are not logically independent then f and g must share at least one common input variable.

Note: It can be seen from the above definition of f and g that logic independence is a *functional* notion and does not use any information about the statistics of the inputs.

For boolean functions, we give the following property:

Proposition 10: Let f and g be two boolean functions and f^c, g^c the cofactors of f and g with respect to a common variable c; if $f^c \perp g^c$ and the variables in their support sets are independent, then f and g are conditionally independent with respect to c that is,

$$p(f \cdot g|c) = p(f|c) \cdot p(g|c) \quad (33)$$

Proof: Shannon's decomposition for f and g gives $f = c\, f^{c} + \bar{c}\, f^{\bar{c}}$, $g = c\, g^{c} + \bar{c}\, g^{\bar{c}}$ respectively; consequently, $f\,g = c\, f^{c}\, g^{c} + \bar{c}\, f^{\bar{c}}\, g^{\bar{c}}$. One can calculate $p\,(f\,g \mid c)$ as:

$p\,(f\,g \mid c) = p\,(f\,g\,c) \,/\, p\,(c) = p\,(c\, f^{c}\, g^{c}) \,/\, p\,(c) = p\,(f^{c}\, g^{c}) = p\,(f^{c})p\,(g^{c})$

because $Sup\,(f^{c}) \cap Sup\,(g^{c}) = \varnothing$. We have also:

$p\,(f \mid c) = p\,(f\,c) \,/\, p\,(c) = p\,(c\, f^{c}) \,/\, p\,(c) = p\,(f^{c})$

$p\,(g \mid c) = p\,(g\,c) \,/\, p\,(c) = p\,(c\, g^{c}) \,/\, p\,(c) = p\,(g^{c})$

therefore $p\,(f \mid c)\, p\,(g \mid c) = p\,(f^{c})p\,(g^{c})$ and this concludes our demonstration. ■

Example: In Fig.10, signals a, b are conditionally independent with respect to c; indeed, we have:

$p\,(a\,b \mid c) = p\,(a\,b\,c) \,/\, p\,(c) = p\,(x\,c\,y\,c\,c) \,/\, p\,(c) = p\,(x\,y\,c) \,/\, p\,(c) = p\,(x)\, p\,(y)$

$p\,(a \mid c)\, p\,(b \mid c) = p\,(a\,c)\, p\,(b\,c) \,/\, p^2(c) = p\,(x\,c)\, p\,(y\,c) \,/\, p^2(c) = p\,(x)\, p\,(y)$

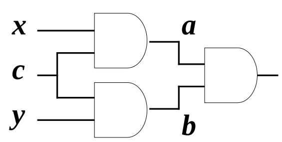


Fig.10: An example to illustrate conditional independence

It's worthwhile to note that, in order to compute $p\,(a\;b\;c)$, if a and b are conditionally independent with respect to c, we may use only pairwise signal probabilities; as we may deduce by simple manipulations:

$p\,(a\,b\,c) = p\,(a\,b \mid c)\, p\,(c) = p\,(a \mid c)\, p\,(b \mid c)\, p\,(c) = p\,(a\,c)\, p\,(b\,c) \,/\, p\,(c)$

which reduces the problem of evaluating the probability of three correlated signals to the one of considering only pairwise correlated signals (if the hypothesis of conditional independence is satisfied).

Consequently, the conditional independence concept can lead to efficient computations even in very complex situations. In fact, Proposition 10 gives us a *sufficient* condition for conditional independence and this is very useful from a practical point of view, because all events appearing in digital logic are somehow logically correlated. However, the general problem, to determine a variable x_i from a set of n signals $\{x_1, x_2,..., x_n\}$ such that the remaining set of $(n - 1)$ signals is conditionally independent with respect to x_i is a complex problem; we will prove in the following that it is actually an **NP**-complete problem.

Proposition 11: (Conditional Independence Problem - CIP)

Given a set of n boolean functions $\{x_1,x_2,...,x_n\}$, an index i and $k \le n - 1$, deciding whether there are at least k signals from the remaining subset conditionally independent with respect to x_i, is an **NP**-complete problem.

Proof: First, we have to prove that CIP is in **NP**. Indeed, given a particular instance of the problem, it may be verified in polynomial time whether the requirements are met.

We will prove that CIP is **NP**-complete using a reduction from the Set Packing Problem [16]:

*Given a collection C of finite sets {S_1, S_2,..., S_n}, a positive integer $k \le |C|$, deciding whether C contains at least k mutually disjoint sets is **NP**-complete.*

Let C and k be as above, $n = |C| + 1$ and x be a boolean function such that $Sup(x) \cap S_j = \varnothing$ for every $j = 1, 2,..., n - 1$ where $S_j \in C$. We build the following boolean functions x_j, $j = 1, 2,..., n - 1$:

$$x_j = x f_j + \bar{x} g_j$$

where f_j and g_j are boolean functions such that $Sup(f_j) = S_j$ and g_j is an arbitrary boolean function. We can see that there is a subset of at least k signals from $x_1, x_2,..., x_{n-1}$ which are conditionally independent with respect to x iff there exists a permutation $i_1, i_2,..., i_{n-1}$ of $1, 2,..., n - 1$ such that the following is true:

$$p\left(\bigcap_{j=1}^{K} x_{i_j} \middle| x\right) = \prod_{j=1}^{K} p(x_{i_j} | x) \quad \text{where } K \ge k.$$

Using the definition of conditional probability and the expression of x_j, we get $\dfrac{p\left(x \bigcap_{j=1}^{K} f_{i_j}\right)}{p(x)} = \prod_{j=1}^{K} \dfrac{p(x f_{i_j})}{p(x)}$.

The construction of x_j's was done such that x and f_j's have disjoint supports so, according to Definition 10, they are logically independent, and thus equivalently we get $p\left(\bigcap_{j=1}^{K} f_{i_j}\right) = \prod_{j=1}^{K} p(f_{i_j})$ which is true iff f_{i_j} are logically independent i.e. their supports are mutually disjoint: $Sup(f_{i_j}) \cap Sup(f_{i_l}) = \varnothing$ or $S_{i_j} \cap S_{i_l} = \varnothing$ for any $j, l \le K$

Thus, the set of signals built above has at least k signals conditionally independent with respect to x iff C has at least k mutually disjoint sets. To conclude, CIP is **NP**-complete.■

One may extend the notion of conditional independence with respect to a single signal to that with respect to a subset of signals. The disadvantage is that, even if we find such a set, we may not express the probability of complex events in terms of probabilities of pairs of events as it is the case with conditional independence with respect to a single signal. Thus, from a computational point of view, this does not seem to be useful. In the following, we will use instead an approximation of conditional independence which holds for correlated inputs.

Definition 10: (Signal Isotropy)

Given the set of n signals $\{x_1, x_2,..., x_n\}$, we say that the conditional independence relation is *isotropic*, if it is true for all signals x_1, x_2,..., x_n; more precisely, taking out all x_i's one at a time, the subset of the remaining (n - 1) signals is conditionally independent with respect to the taken x_i.

Returning to our example in Fig.10, given the set of signals $\{a, b, c\}$ we have that $\{a, b\}$ is conditionally independent with respect to c, but the sets $\{a, c\}$ or $\{b, c\}$ are not conditionally independent with respect to b, or a, respectively; it follows that conditional independence is not isotropic in this particular case. Intuitively, the concept of isotropy as defined above, is restrictive by its very nature and it is hardly conceivable that a set of signals taken randomly from a target circuit will satisfy Definition 10. Our goal, however, is not to use this concept as it is, but to make it more practical for our purposes. As we shall see later, the main advantage of isotropy is that it offers a canonical approach to the estimation of different kinds of probabilities in digital circuits.

Definition 11: (Almost Isotropy)

The property of conditional independence for a set of n signals $\{x_j\}_{1 \le j \le n}$ is called *almost isotropic* if there exists some ε (ε ≥ 0) so that it is satisfied within ε relative error for any possible permutation of signals x_i:

$$\left| \frac{\prod_{1 \le j \le n,\ j \ne i} p(x_j | x_i)}{p\left(\bigcap_{1 \le j \le n,\ j \ne i} x_j \middle| x_i\right)} - 1 \right| \le \varepsilon \quad \text{for any } i = 1, 2, ..., n \tag{34}$$

Differently stated, almost isotropy is an approximation of isotropy within given bounds of relative error. A natural question may arise now: how often it is appropriate to consider almost isotropy as an approximation of pure isotropy? To answer this question, we consider in Fig.11 several common situations involving the set of signals $\{u, v, w\}$ and the relative position of their logic cones (each cone illustrates the dependence of signals u, v, w on the primary inputs). Whilst the isotropy is completely satisfied only in (b), the almost isotropy concept is applicable in all other cases; more precisely, the conditional independence relation is partially satisfied in (a) with respect to w, in (c) with respect to u and v and in (d) with respect to u and v.

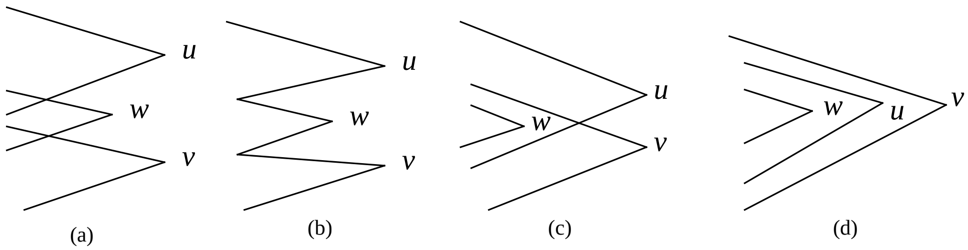


Fig.11: An example to illustrate pure and almost isotropy

Based on the previous definition, we get the following:

Proposition 12: Given an almost isotropic set of signals $\{x_j\}_{1 \le j \le n}$ for some ε, the probability of the composed signal $p(\bigcap_{j=1}^{n} x_j)$ may be estimated within ε relative error as:

$$p\left(\bigcap_{j=1}^{n} x_j\right) = \frac{\left(\prod_{1 \le i < j \le n} p(x_i x_j)\right)^{\frac{2}{n}}}{\left(\prod_{i=1}^{n} p(x_i)\right)^{\frac{n-2}{n}}} \qquad (35)$$

Proof: From the definition of almost isotropy, we get the following: for every $i = 1, 2,..., n$:

$\left| \frac{\prod_{1 \le j \le n,\ j \ne i} p(x_j | x_i)}{p\left(\bigcap_{1 \le j \le n,\ j \ne i} x_j \middle| x_i\right)} - 1 \right| \le \varepsilon$ which may be re-written using the definition of conditional probability as:

$$1 - \varepsilon \le \frac{\dfrac{\prod_{1 \le j \le n,\ j \ne i} p(x_i x_j)}{p^{n-2}(x_i)}}{p\left(\bigcap_{j=1}^{n} x_j\right)} \le 1 + \varepsilon \qquad \text{for each } i = 1, 2,..., n$$

Multiplying all inequalities we get exactly the above claim. ■

This proposition provides us a very strong result: given that *n* signals are almost isotropic for some ε, the probability of their conjunction may be estimated within ε relative error using only the probabilities of pairs of signals, thus reducing the problem complexity from exponential to quadratic. This is similar to the concept of pairwise correlation coefficient introduced in [8] and generalized in [3]. However, the approach in [3] does not provide sufficient accuracy for highly correlated signals as we shall see later.

5.3. An Incremental Propagation Mechanism Using Almost Isotropy

If the almost isotropy property is satisfied, Proposition 12 may be easily extended to boolean functions represented by OBDDs. Let *f* be a boolean function of *n* variables $x_1, x_2, \ldots, x_n$ which may be defined through the ON- and OFF-sets as in Section 4. Based on this representation, we give the following result:

Proposition 13: Given *f* a boolean function of variables $x_1, x_2, \ldots, x_n$, the following hold:

a) If the set $\{x_j\}_{1 \le j \le n}$ in which each variable is either direct or complemented is almost isotropic for some ε (ε ≥ 0), then the *signal probability* $p(f = i)$ with $i = 0, 1$ may be expressed within ε relative error as:

$$p(x = i) = \sum_{\pi \in \Pi_i} \frac{\left(\prod_{1 \le k < l \le n} p(x_k = i_k \cap x_l = i_l)\right)^{\frac{2}{n}}}{\left(\prod_{k=1}^{n} p(x_k = i_k)\right)^{\frac{n-2}{n}}} \tag{36}$$

where i_k is the value taken by variable x_k in the cube $\pi \in \Pi_i$.

b) If the set $\{x_{j_{k \to l}}\}_{1 \le j \le n,\, k,\, l = 0,\, 1}$ is almost isotropic for some ε (ε ≥ 0), then the *transition probability* $p(f_{i \to j})$ with $i, j = 0, 1$ may be expressed within ε relative error as:

$$p(f_{i \to j}) = \sum_{\pi \in \Pi_i} \sum_{\pi' \in \Pi_j} \frac{\left(\prod_{1 \le k < l \le n} p(x_{k_{i_k \to j_k}} \cap x_{l_{i_l \to j_l}})\right)^{\frac{2}{n}}}{\left(\prod_{k=1}^{n} p(x_{k_{i_k \to j_k}})\right)^{\frac{n-2}{n}}} \tag{37}$$

where i_k, j_k are the values taken by the variable x_k in the cubes $\pi \in \Pi_i$. and $\pi' \in \Pi_j$.

Proof: a) We may define the event '*f* takes the value *i*' as:

$$(f = i) = \bigcup_{\pi \in \Pi_i} \bigcap_{k=1}^{n} (x_k = i_k)$$

In the probabilistic domain, this becomes:

$$p(f = i) = \sum_{\pi \in \Pi_i} p\left(\bigcap_{k=1}^{n} (x_k = i_k)\right)$$

because the paths $\pi \in \Pi_i$ are disjoint. Since the input variables are almost isotropic for some ε (0 ≤ ε < 1), we may use Proposition 12 to express the probability of each cube, getting exactly (36).

b) Proof similar to case a) but considering instead the event '*f* switches from *i* to *j*' expressed as:

$$f_{i \to j} = \bigcup_{\pi \in \Pi_i} \bigcup_{\pi' \in \Pi_j} \bigcap_{k=1}^{n} x_{k_{i_k \to j_k}} . \blacksquare$$

The above result may be reformulated using signal and transition correlation coefficients; it may be used in signal probability and switching activity estimation, given that the almost isotropy conditions are met.

Corollary 1: Given a set of signals $\{x_j\}_{1 \le j \le n}$ as in Proposition 13 and a boolean function *f* of variables $\{x_j\}_{1 \le j \le n}$, the following hold within ε relative error:

$$\text{a)}\ p(f = i) = \sum_{\pi \in \Pi_i} \left(\prod_{1 \le k < l \le n} SC_{i_k i_l}^{x_k x_l} \right)^{\frac{2}{n}} \prod_{k=1}^{n} p(x_k = i_k) \tag{38}$$

$$\text{b)}\ p(f_{i \to j}) = \sum_{\pi \in \Pi_i} \sum_{\pi' \in \Pi_j} \left(\prod_{1 \le k < l \le n} TC_{i_k i_l,\, j_k j_l}^{x_k x_l} \right)^{\frac{2}{n}} \prod_{k=1}^{n} p(x_{k_{i_k \to j_k}}) \tag{39}$$

Proof: a) and b) Using the definition of transition probability and Proposition 13, we easily get the above inequalities.■

This result can also be extended to the calculation of correlation coefficients (*SCs* or *TCs*) between two signals in the circuit. From a practical point of view, this becomes an important piece in the propagation mechanism of probabilities and coefficients through the boolean network. We get the following:

Proposition 14: Given a set of signals $\{x_j\}_{1 \le j \le n}$, a boolean function *f* of variables $\{x_j\}_{1 \le j \le n}$, and *x* a signal from the circuit, if $\{x_1, x_2, ..., x_n, x\}$ is a set as in Proposition 13, then the correlation coefficients (*SCs* and *TCs*) can be expressed within ε relative error[1] as:

$$\text{a)}\ SC_{ij}^{fx} = \frac{\sum_{\pi \in \Pi_i} \left(\prod_{1 \le k < l \le n} SC_{i_k i_l}^{x_k x_l} \right)^{\frac{2}{n+1}} \prod_{k=1}^{n} \left(p(x_k = i_k) (SC_{i_k j}^{x_k x})^{\frac{2}{n+1}} \right)}{p(f = i)} \tag{40}$$

$$\text{b)}\ TC_{ip,\, jq}^{fx} = \frac{\sum_{\pi \in \Pi_i} \sum_{\pi' \in \Pi_j} \left(\prod_{1 \le k < l \le n} TC_{i_k i_l,\, j_k j_l}^{x_k x_l} \right)^{\frac{2}{n+1}} \prod_{k=1}^{n} \left(p(x_{k_{i_k \to j_k}}) (TC_{i_k p,\, j_k q}^{x_k x})^{\frac{2}{n+1}} \right)}{p(f_{i \to j})}$$

where *i, j, p, q* = 0, 1.

Proof: a) and b) follow directly from the definition of *SC*s and *TCs* and using the events '*f* = *i* and *x* = *j* simultaneous' and '*f* switches from *i* to *j* and *x* from *p* to *q* simultaneously', respectively.■

1. This ε is the maximum over all values that occur during the incremental propagation process.

These results lead to a new heuristic algorithm for signal and transition probability estimation under input streams which exhibit spatiotemporal correlations. We may thus see equation (39) as the improvement of (25) by using the notions of conditional independence and signal isotropy. Compared to the heuristic proposed in Section 4, this new approach based on conditional independence has also the advantage that it supplies bounds for error estimation provided the input signals are almost isotropic. This bounding value could not be provided using the spatiotemporal hypothesis alone. Finally, the model introduced above provides a way to improve the run time requirement as shown below.

Proposition 15: If C_j is a correlation coefficient (*SC* or *TC*) at level *j* (given by a topological order from inputs to outputs of the circuit), then it is related to C_{j-l} ($0 < l < j$) by a proportionality relationship expressible as $C_l \propto (C_{j-l})^{\left(\frac{2}{n+1}\right)^l}$ where *n* represents the average fan-in value in the circuit.

Proof: Follows from Proposition 14 and Corollary 1 if the conditions required in Proposition 13 are satisfied.■

Corollary 2: If $l \to \infty$ then the signals behave as uncorrelated.

Proof: Follows immediately from Proposition 15; more precisely, $(C_{j-l})^{\left(\frac{2}{n+1}\right)^l} \to 1$ when $l \to \infty$ and that represents the condition of noncorrelation in our approach.■

In other words, we do not need to compute the coefficients which are beyond some level *l* in the circuit; instead, we may assume them equal to 1 without decreasing the level of accuracy. Also, *the larger the average fanin n of the circuit, the smaller value for l* may be used. It is worthwhile to note that the conditional independence relationship, more specifically the almost isotropy, is essential for this conclusion. The approach based on spatiotemporal correlations *only*, does not provide a sufficient rationale for such a limitation. This is actually a very important heuristic to use in practice and its impact on run time is huge; limiting the number of calculations for each node in the boolean network to a fixed amount (which depends on the value set as threshold for *l*) reduces the problem of coefficients estimation from *quadratic* to *linear* complexity.

6. Practical considerations and experimental results

All experiments were performed using SIS environment on an Ultra SPARC 2 workstation with 64Mbytes of memory; the working procedure is shown below:

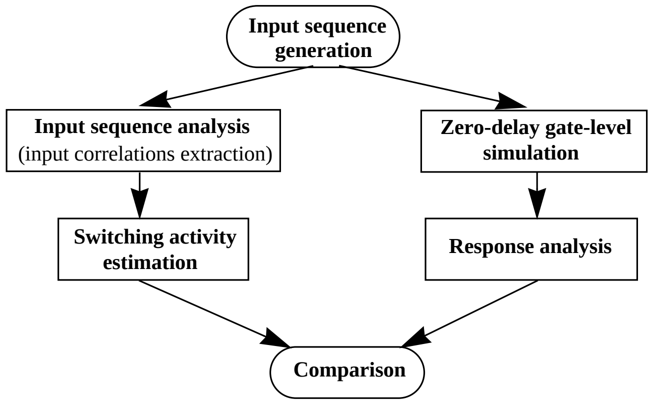


Fig.12: The experimental setup

To generate pseudorandom inputs we have used as input generator a maximal-length linear feed-back shift register modified to include the all-zero pattern [17]; these registers are based on primitive polynomials that is, they randomly generate all distinct patterns that correspond to a given polynomial before repeating the sequence. Purely random generators do not exist, therefore the primitive polynomials used, give us multiple correlations among primary inputs. The length of the input register was set equal to the number of inputs of the circuit under analysis, thereby creating a pseudorandom source; when the length of this register became huge, we tried to keep the time/space requirements at a reasonable level and hence, for these cases we generated only a significant part of the exhaustive sequence (up to 2^{20} input patterns).

As the standard measure for power estimation, we have used the average switching activity at each node of the circuit calculated as in equation (10). In our experiments, we were mainly interested in measuring the accuracy of the model in estimating the switching activity locally (at each internal node of interest) and globally (for the entire circuit), given a set of inputs with spatiotemporal correlations. The analysis part of the experiment may be skipped if the user specifies directly the characteristics of the input stream (transition probabilities and correlation coefficients).

To illustrate the main features of our approach, we consider in Fig.13 the ISCAS circuit C17, fed by the sequence generated with the primitive polynomial $p(x) = 1 \oplus x \oplus x^3$. Due to the deterministic way in which we generate the input sequence, independent lines become correlated as is the case with inputs 1 & 2, 2 & 3, 3 & 6, 6 & 7; in turn, the fan-out points on the input lines add additional correlations. For an accurate analysis of the switching activity, we have to account for all these dependencies. In Table 1 (a) we list the transition probability coefficients for this particular input sequence; we mention that in this case, all signal correlation coefficients are equal to 1, therefore they cannot make a difference alone. In Table 1 (b) we present the estimated and exact values of the switching activity per clock cycle. In Fig.13, the color code is used to reflect the switching activity at the output of the gates, i.e. darker gates are more active (in particular, gate 23 is the most active one).

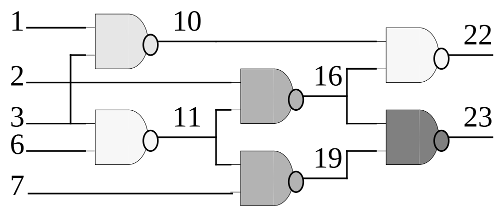


Fig.13: Switching activities distribution in C17 (pseudorandom inputs)

Table 1: (a) C17: TCs for pseudorandom inputs

ij,kl	1&2	2&3	3&6	6&7
00,00	2.0000	2.0000	2.0000	1.7143
00,01	0.0000	0.0000	0.0000	0.4444
00,10	2.0000	2.0000	2.0000	1.7778
00,11	0.0000	0.0000	0.0000	0.0000
01,00	2.0000	2.0000	2.0000	2.2857
01,01	0.0000	0.0000	0.0000	0.0000
01,10	2.0000	2.0000	2.0000	1.7778
01,11	0.0000	0.0000	0.0000	0.0000
10,00	0.0000	0.0000	0.0000	0.0000
10,01	2.0000	2.0000	2.0000	1.7778
10,10	0.0000	0.0000	0.0000	0.4444
10,11	2.0000	2.0000	2.0000	1.7143
11,00	0.0000	0.0000	0.0000	0.0000
11,01	2.0000	2.0000	2.0000	1.7778
11,10	0.0000	0.0000	0.0000	0.0000
11,11	2.0000	2.0000	2.0000	2.2857

Table 1: (b) C17: sw_act for pseudorandom inputs

Node	Estimated sw_act	Exact sw_act
1	0.5000	0.5000
2	0.5000	0.5000
3	0.5000	0.5000
6	0.5000	0.5000
7	0.5625	0.5625
10	0.3750	0.3907
11	0.2500	0.2649
16	0.5000	0.5236
19	0.5687	0.5236
22	0.2978	0.3125
23	0.6006	0.5625

It should be pointed out that the actual values for all coefficients in Table 1 (a) represent the characteristics of the input stream; if we select another primitive polynomial to generate the inputs, we may obtain a completely different set of transition correlation coefficients. This dependency is even more salient if we consider ‘biased inputs’ (i.e. the switching activity is not 0.5). To generate such sequences, we used a simple functional generator based on the *random* function in C language. For each bit, we set up a specific threshold $t \in [0,1]$ and generated a set of random numbers in [0,1]. If these numbers exceeded the threshold t, then the output of the generator is set to 1; otherwise the output is 0. We give in Tables 2(a) ($t = 0.25$ for all bits) and 2(b) ($t \in [0.1, 0.5]$), the values obtained for two such sequences, and in Fig.14 the new distribution of switching activity among the internal nodes of the circuit. As we can see, gates 22 and 19 become in turn the most active gates in these two experiments. We note that in these cases we have spatial dependencies among all primary inputs.

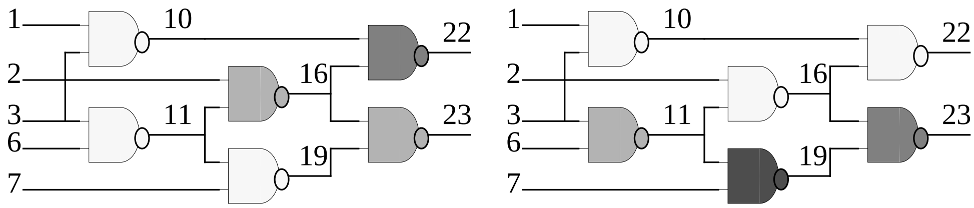


Fig.14: Switching activities distribution in C17 for two biased input sequences

Table 2: (a) C17: sw_act for biased inputs

Node	Estimated sw_act	Exact sw_act
1	0.5625	0.5625
2	0.3750	0.3750
3	0.3125	0.3125
6	0.5625	0.5625
7	0.5000	0.5000
10	0.3125	0.3125
11	0.3125	0.3125
16	0.4569	0.5000
19	0.3367	0.3125
22	0.4960	0.5000
23	0.4527	0.4375

Table 2: (b) C17: sw_act for biased inputs

Node	Estimated sw_act	Exact sw_act
1	0.0625	0.0625
2	0.1250	0.1250
3	0.2500	0.2500
6	0.5000	0.5000
7	1.0000	1.0000
10	0.1250	0.1250
11	0.2500	0.2500
16	0.1295	0.1250
19	0.7262	0.7500
22	0.2040	0.1250
23	0.4597	0.5000

To further assess the impact of correlations, we considered the benchmark *f51m* and the following two scenarios:

a) *Low Correlations:* the input patterns are generated by a linear feedback shift register which implements the primitive polynomial: $p(x) = 1 \oplus x \oplus x^2 \oplus x^7 \oplus x^8$;

b) *High Correlations:* the input patterns are generated using the state lines of an 8-bit counter.

In order to do a fair comparison between the existing estimation techniques (including the ones which use global OBDDs) and our technique, we had to choose a small sized circuit.

The estimated values in both cases were compared against the exact values of switching activity obtained by exhaustive simulation; all internal nodes and primary outputs have been taken into consideration (see Fig. 15)[1].

1. In Fig.15, on the *x* axis, we report the absolute error of switching activity that is, $|sw_{exact} - sw_{estimated}|$.

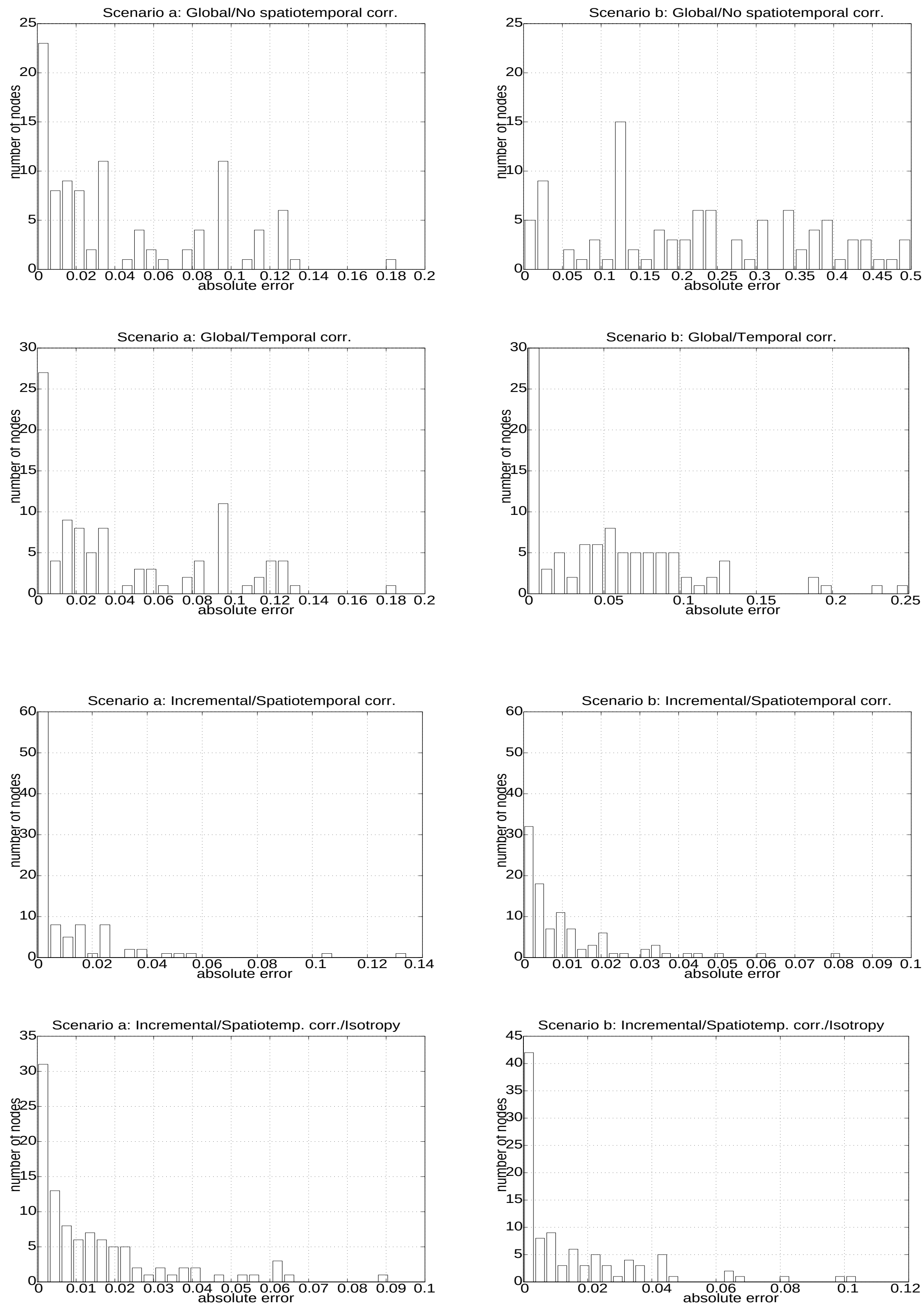


Fig.15: The impact of the level of correlation on switching activity estimation in *f51m*

In Scenario (a), all approaches are quite accurate. However, we point that only considering spatiotemporal correlations and signal isotropy ensures the highest accuracy (100% of the nodes estimated with error less than 0.1). As the results of Scenario (b) show, the level of correlation on the primary inputs strongly impacts the quality of estimation. Specifically, it makes completely inaccurate the global approach based on input independence (despite the fact that internal dependencies due to reconvergent fan-out are accounted by building the global OBDD). As expected, this is visible mostly in Scenario (b), where less than 20% of the nodes are estimated with precision higher than 0.1. On the other hand, even if temporal correlations are taken into account, but the inputs are assumed to be spatially uncorrelated, only 80% of the nodes are estimated with error less than 0.1. Accounting for spatiotemporal correlations provides excellent results for highly correlated inputs (100% nodes estimated with precision 0.1), but the mean error in the hypothesis of conditional independence is anyway smaller (90% of the nodes are estimated with error less than 0.05). This results clearly demonstrate that power estimation is a strongly pattern dependent problem, therefore accounting for dependencies (at the primary inputs and internally, among the different signal lines) is mandatory if accuracy is important. From this perspective, accounting for spatiotemporal correlations and using the conditional independence and signal isotropy concepts seems to be the best candidate to date.

Using some ISCAS'85 benchmarks, we further performed the following types of experiments:

a) one set of experiments to validate the model based only on spatiotemporal correlations without conditional independence;

b) another one to assess the impact of the limiting technique from Proposition 15;

c) the last one to validate the model based on spatiotemporal correlations with conditional independence and signal isotropy.

Once again, the switching activity values and power consumption were estimated at *each* internal node and primary output and compared with the ones obtained from logic simulation. We found that power estimation for the entire circuit is not a real measure to use in low-power design and power optimization where the switching activity at *each* node has to be accurately estimated.

a) Experiments to validate the model based only on spatiotemporal correlations

In the following, we give the error values for pseudorandom inputs, using the incremental approach without conditional independence. In reporting the error, we compared our switching activity estimates with the results of logic simulation at every internal node and primary output. All benchmarks were mapped with *lib2genlib*.

Table 3: Pseudorandom inputs with no limit[a] ($l \to \infty$) in *TC*s calculation

Circuit	MAX	MEAN	RMS	STD	TIME (sec.)
C17	0.0565	0.0119	0.0238	0.0226	0.04
C432	0.0716	0.0133	0.0222	0.0179	30.17
C499	0.0334	0.0047	0.0072	0.0055	88.17
C880	0.1131	0.0158	0.0306	0.0264	45.38
C1355	0.0393	0.0027	0.0051	0.0044	61.76
C1908	0.0353	0.0044	0.0082	0.0069	67.15
C3540	0.1765	0.0276	0.0429	0.0318	491.19
C6288	0.2046	0.0204	0.0443	0.0396	616.22

a. l is the limit used in Proposition 15 and Corollary 2.

b) Experiments concerning run time improvement

Heuristic proposed in Section 5.3 is important in practice not only for substantially reducing the run time but also for keeping the same level of accuracy as the case when the threshold limit is set to infinity (that is we didn't use any limitation in *TC*s calculation). In the following, we present a detailed analysis for the benchmark *duke2* which exhibits a typical behavior; in the first case the limit was set to infinity, in the second one the limit was 4. To report error, we used standard measures for accuracy: maximum error (MAX), mean error (MEAN), root-mean square (RMS) and standard deviation (STD); we excluded deliberately the relative error from this picture, due to its misleading prognostic for small values.

Table 4: *duke2* - Speed-up vs. accuracy

	LOW CORRELATIONS		HIGH CORRELATIONS	
Error	NO LIMIT ($l \to \infty$)	LIMIT $l = 4$	NO LIMIT ($l \to \infty$)	LIMIT $l = 4$
MAX	0.0744	0.0710	0.0299	0.0299
MEAN	0.0133	0.0161	0.0056	0.0055
RMS	0.0223	0.0269	0.0085	0.0083
STD	0.0182	0.0219	0.0065	0.0063
TIME	131.08 s	40.83 s	130.55 s	42.14 s

As we can see, the quality of estimation is practically the same in both cases whilst the run time is significantly reduced in the second approach. It should be pointed out, that this limiting technique works fine also for partitioned circuits which is an essential feature in hierarchical analysis for power estimation. Running extensively our estimation tool on circuits of various sizes and types (ISCAS'85 benchmarks, adders, multipliers), we observed the following general tendency for speed-up:

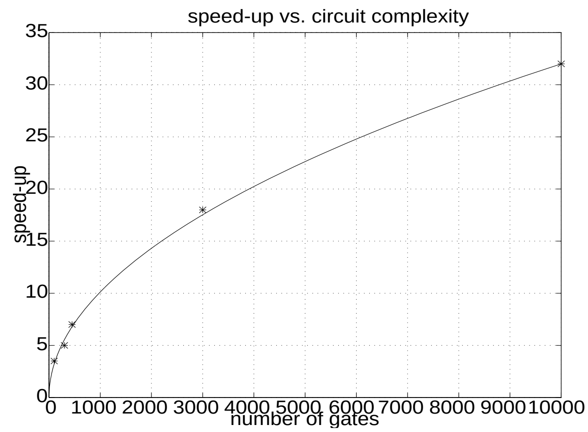


Fig. 16: Speed-up factor vs. circuit complexity

We can see that the speed-up is about 3 ÷ 5 times for less complex circuits, but it may become 20 ÷ 30 times for large examples; we estimated the power consumption for multipliers on 16 bits (2048 gates) and 32 bits (9124 gates) and the run times were 43.90 and 195.13 sec.

We present in Table 5 the results obtained for the set of ISCAS'85 benchmarks using the limit $l = 4$ in *TC*s calculation. By comparing these results with those given in Table 3, we can see that the quality of the estimates remained basically the same while the run time was significantly improved.

Table 5: Pseudorandom inputs with limit $l = 4$ in *TC*s calculation

Circuit	MAX	MEAN	RMS	STD	Total Power	TIME (sec.)
C432	0.1916	0.0281	0.0465	0.0374	3372.57	11.82
C499	0.0624	0.0134	0.0184	0.0126	7645.56	10.57
C880	0.0691	0.0135	0.0211	0.0164	6391.83	18.35
C1355	0.0225	0.0041	0.0051	0.0030	6797.92	4.24
C1908	0.1315	0.0091	0.0206	0.0185	7435.34	12.69
C3540	0.2010	0.0307	0.0509	0.0407	16356.82	60.86
C6288	0.0890	0.0142	0.0241	0.0196	46846.48	29.10

c) Experiments to validate the conditional independence and signal isotropy

The experiments were performed on large ISCAS'85 examples using pseudorandom and highly correlated inputs (obtained from counted sequences of length 2^{20}); all results reported here, have been derived using the value $l = 4$ as the limit for coefficients calculations. To report error, all estimations were verified against exhaustive simulation performed with SIS logic simulator. To show the impact of using signal isotropy concept, for high-correlation scenario, we also present in Table 6 the results obtained if signal isotropy is not used.

Table 6: High-correlations on primary inputs

	WITH conditional independence (ε-isotropy)					WITHOUT conditional independence				
Circuit	MAX	MEAN	RMS	STD	Total power	MAX	MEAN	RMS	STD	Total power
C432	0.2538	0.0225	0.0585	0.0545	306.88	0.8499	0.1058	0.2274	0.2032	1390.07
C499	0.1566	0.0421	0.0760	0.0634	2283.03	0.4254	0.0387	0.0933	0.0851	2543.99
C880	0.0175	0.0013	0.0040	0.0038	263.13	0.7853	0.0471	0.1630	0.1571	482.60
C1355	0.1930	0.0227	0.0520	0.0469	1865.81	0.4722	0.0516	0.1252	0.1144	1961.76
C1908	0.3907	0.0294	0.0868	0.0820	3156.83	0.4903	0.0459	0.1000	0.0892	3201.10
C3540	0.0279	0.0279	0.0030	0.0030	166.25	0.5463	0.0280	0.0365	0.0365	207.38
C6288	0.1773	0.0231	0.0521	0.0471	8843.72	0.5639	0.1092	0.1995	0.1685	19428.91

As we can see, by using conditional independence and signal isotropy, the accuracy for a node-by-node analysis improves on average by an order of magnitude; on the other hand, by not using conditional independence at all, the total power consumption is overestimated by 100% on average.

To further asses the impact of the correlation level, we considered the following experiment. We estimated the total power consumption for a set of ISCAS circuits using random inputs (the low correlation scenario in Table 7). After that, we sorted the input vectors (therefore we kept the same signal probability on the inputs) and we applied the resulting file to the same set of circuits (the medium correlation scenario). Finally, we considered as input a counted sequence with the same average number of transitions (but different signal probability) and we estimated once again the total power consumption (the high correlation scenario in Table 7). All the values of power consumption are reported in μW at 20 MHz.

Table 7: Total Power Consumption (μW @ 20Mhz)

Circuit	Low Correlations	Medium Correlations	High Correlations
Circuit 1: C432	3372.57	2046.93	306.88
Circuit 2: C499	7645.56	5068.32	2283.03
Circuit 3: C880	6391.83	4781.47	263.13
Circuit 4: C1355	6797.92	4272.74	1865.81
Circuit 5: C1908	7435.34	4307.76	3156.83
Circuit 6: C3540	16356.82	8464.05	166.25
Circuit 7: C6288	46846.48	40748.45	8843.72

As we can see, there is a significant difference in all cases: not only the switching activities at each internal node were completely different as the level of inputs correlation changes, but also the values of total power consumption. For example, for C432, the total power estimated under low correlated inputs was 3372.57 μW, while this value for strongly correlated inputs was 306.88 μW (there is a factor of 11 between the two). The same behavior has been observed for other circuits. To give a more meaningful measure to this experiment, we give in Fig.17 the error made in medium and high correlation scenarios by ignoring the correlations on the circuit inputs (o-medium correlations and *-high correlations).

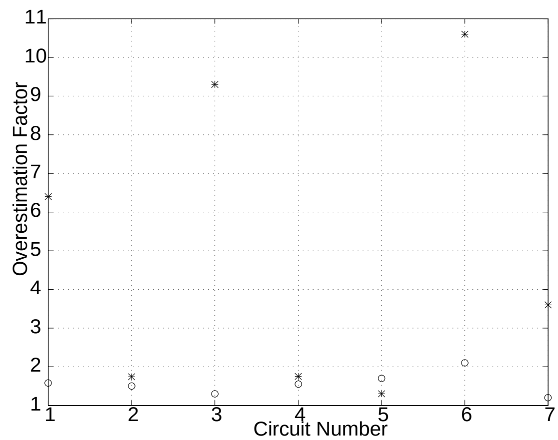


Fig.17: Overestimation factor by ignoring input correlations

As we can see, once again the level of correlations has a significant impact on our predictions; whilst for medium correlations one may overestimate in general about twice, for high correlations the overestimate is about 10 (e.g. Circuit 6: C3540, Circuit3: C880).

To conclude, input pattern dependencies (in particular highly correlated inputs) is an extremely important issue in power estimation. From this perspective, power analysis needs analytical models to overcome this difficulty.

7. Conclusion

In this paper, we have proposed an original approach for switching activity estimation in combinational logic modules under pseudorandom or highly biased input streams. Using the zero-delay hypothesis, we have derived a probabilistic model based on lag-one Markov Chains which supports spatiotemporal correlations among the primary inputs and internal lines of the circuit under consideration. The main feature of our approach is the systematic way in which we can deal with complex dependencies that may appear in practice. From this perspective, the new concepts of conditional independence and signal isotropy are used in a uniform manner to fulfill practical requirements for fast and accurate estimation. Under general assumptions, the conditional independence problem has been shown to be **NP**-complete; consequently, efficient heuristics have been provided for probabilities and coefficients calculation. As a distinctive feature, this approach improves the state-of-the-art in two ways: theoretically by providing a deep insight about the relationship between the logical and probabilistic domains, and practically by offering a sound mathematical framework and an efficient technique for power analysis.

References

[1] N. Weste, K. Eshraghian, ‘Principles of CMOS VLSI Design: A Systems Perspective’, Addison Wesley Publishing Company, 1988.

[2] J. Rabaey, and M. Pedram, ‘Low Power Design Methodologies’, Kluwer Academic Publishers, 1996.

[3] R. Marculescu, D. Marculescu, and M. Pedram, ‘Switching Activity Analysis Considering Spatiotemporal Correlations’, in *Proc. IEEE/ACM Intl. Conference on Computer Aided Design*, pp. 294-299, Nov.1994.

[4] R. Marculescu, D. Marculescu, and M. Pedram, ‘Efficient Power Estimation for Highly Correlated Input Streams’, in *Proc. ACM/IEEE Design Automation Conference*, pp. 628-634, June 1995.

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

[6] L.H. Goldstein, ‘Controllability/Observability Analysis of Digital Circuits’, in *IEEE Trans. Circuits and Systems*, vol. CAS-26, pp. 685-693, Sept.1979.

[7] J. Savir, G. S. Ditlow, and P. H. Bardell, ‘Random Pattern Testability’, in *IEEE Trans. on Computers*, vol. C-33, pp. 1041-1045, Jan.1984.

[8] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, ‘Testability Measures in Pseudorandom Testing’, in *IEEE Trans. on CAD*, vol. 11, pp. 794-800, June 1992.

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

[10] F. N. Najm, R. Burch, P. Yang, and I. Hajj, ‘Probabilistic Simulation for Reliability Analysis of CMOS VLSI Circuits’, in *IEEE Trans. on CAD*, vol. CAD-9, pp. 439-450, April 1990.

[11] A. Ghosh, S. Devadas, K. Keutzer, and J. White, ‘Estimation of Average Switching Activity in Combinational and Sequential Circuits’, in P*roc. ACM/IEEE Design Automation Conference*, pp. 253-259, June 1992.

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

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

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

[15] A. Papoulis, ‘Probability, Random Variables, and Stochastic Processes’, McGraw-Hill Co., 1984.

[16] M. Garey, and D. Johnson, ‘Computers and Intractability’, New York: Freeman, 1979.

[17] P. H. Bardell, W. H. McAnney, and J. Savir, ‘Built-in Test for VLSI: Pseudorandom Techniques’, J. Wiley & Sons Inc. 1987.