Power Analysis for Sequential Circuits at Logic Level

Peter H. Schneider[1,2], Matthias A. Senn[2], and Bernd Wurth[3]

[1] Siemens AG
ZFE T SE 5
D-81730 Munich

[2] Inst. of EDA
TU-Munich
D-80290 Munich

[3] Synopsys, Inc.
700 E. Middlefield Road
Mountain View, CA, 94043

Abstract - *This paper introduces a novel technique to determine the transition probabilities of internal signals for sequential circuits. We account for temporal correlations of primary inputs and internal signals, sequential correlations, and spatial correlations of internal signals. For this purpose, we exploit and combine concepts of unrolling, reconvergence analysis, decomposing packets of temporally correlated variables, and Markov chains. Experimental results demonstrate the high accuracy and efficiency of our technique.*

1. Introduction

1.1. Power Consumption

For CMOS circuits, power consumption is dominated by charging and discharging capacitances when a transition occurs on a signal [7]. The average power consumption in such a circuit is given by

$$P_{circuit} = \frac{1}{2} V_{dd}^2 f \sum_{signal\, i} C(i)\, E(i)$$

where $C(i)$ is the sum of all capacitances of the transistors that are driven by signal i. The supply voltage is denoted by V_{dd} and the clock frequency by f. $E(i)$ is the *transition probability* at signal i, i.e., the probability that there is a $1 \rightarrow 0$ or a $0 \rightarrow 1$ transition on signal i from one clock cycle to the next.

1.2. Logic Synthesis for Low Power Needs Proper Power Analysis

Logic synthesis transforms a circuit step by step, and each step optimizes with respect to the cost function. One transformation step (e.g., decomposition) changes only a small part of a circuit. The general power minimization strategy at logic level is to decrease $\sum C(i)E(i)$. Therefore, the transition probabilities are parameters of the cost function, and after each optimization step, transition probabilities must be re-estimated. Thus, an estimation of the transition probabilities must be accurate and fast. Accurate estimation is necessary to guide the optimization process. Fast estimation allows to apply a large number of optimization steps and thus also contributes to the design quality.

1.3. Correlations

Estimation accuracy severely depends on whether temporal, spatial, and sequential correlations are taken into account.
- *Temporal correlations:* Signals may be temporally correlated, i.e., the next value of a signal depends on its current value. The example

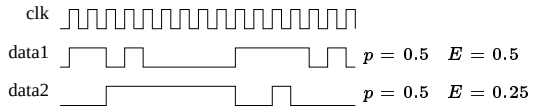

Figure 1. Temporal correlation

in Fig. 1 illustrates temporal correlations. Both, the signal "data1" and the signal "data2" have *signal probability* p equal to 0.5, i.e., both signals take on value 1 with probability $p(\text{data1}) = p(\text{data2}) = 0.5$. Let the signal "data1" be temporally uncorrelated in this example. Then, it switches every second cycle on average, which yields *transition probability* $E(\text{data1}) = 0.5$. Estimation methods that do not consider temporal dependence will assume that the signal "data2" also switches on every second cycle on average. However, the signal "data2" actually switches on every fourth cycle only. Thus, its transition probability is one fourth ($E(\text{data2}) = 0.25$). Neglecting temporal correlations causes major inaccuracies in this example.

- *Spatial correlations:* Fanout signals induce spatial correlations of internal signals even if primary inputs are spatially uncorrelated. The necessity of considering these spatial correlations of internal signals will be shown by experimental results. Furthermore, primary inputs can be spatially correlated for some applications. We assume primary inputs to be spatially uncorrelated.
- *Sequential correlations:* Even if primary inputs are uncorrelated, states and state lines can be temporally correlated. A binary counter with the reset signal as the only input is an example. The reset signal may be uncorrelated and has a very low signal probability. Nevertheless, the state lines are higher order temporally and spatially correlated. All these correlation, which are introduced by the sequential circuit structure, are called sequential correlations.

1.4. State of the Art

To determine transition probabilities in sequential circuits, several techniques have been presented so far. They can be divided into two categories. The first category consists of techniques based on simulation. Simulation based techniques yield high accuracy if a large number of input patterns is applied. Recently published approaches [11, 4] employ the Monte Carlo technique to give confidence measurements on the accuracy of state line probabilities. The disadvantage of simulation is that the complete circuit or at least the transitive fanin must be simulated. As explained in Section 1.2, logic synthesis typically transforms only a small part of a circuit in each optimization step. Reanalysis of the small part is sufficient.

The second category consists of approaches that compute transition probabilities based on probabilistic techniques. Most approaches for sequential circuits assume temporally and spatially uncorrelated primary inputs. Under this assumption, exact methods are presented in [17, 8]. These methods assume an FSM model of a sequential circuit and are based on the Chapman-Kolmogorov equations. This exact technique has problems. First, these linear equations assume that the present state is uncorrelated to the present primary input vector. This does not hold for temporally correlated inputs. In case of temporally correlated inputs, the present input vector and the present state are correlated since both depend on the previous input vector. Thus, for temporally correlated primary inputs, the techniques based on the Chapman-Kolmogorov equations yield inaccurate results. This is illustrated in the appendix. Furthermore, the circuit size that can be handled is limited since the Chapman-Kolmogorov equations are built in terms of state probabilities. This is too expensive for large circuits.

In order to overcome the limitation on circuit size, approximation techniques were suggested in [17]. Contrary to the Chapman-Kolmogorov based techniques, the approximation techniques consider state lines. Two different techniques are suggested to deal with sequential correlations. Both obtain a system of non-linear equations that approximately accounts for correlations of the state lines. The Picard-Peano method or the Newton-Raphson method is applied to solve the system of equations.

After preliminary results of our work had been presented in [15], two techniques to account for temporal correlations at primary inputs have been suggested in [3]. Firstly, the Chapman-Kolmogorov based technique has been extended, but no results have been given. Secondly, the approximation technique presented in [17] has been combined with Markov Chains. Results have been given only for small circuits with less than 20 flipflops. CPU times are 10 - 100 times higher than with our technique.

EURO-DAC '96 with EURO-VHDL '96
0-89791-848-7/96 $4.00

1.5. Our Approach

In this paper, a novel probabilistic approach for sequential circuits is presented that accounts for temporal correlations of primary inputs. To deal with sequential correlations, we think of a circuit to be unrolled as suggested in [6, 17]. Thus, sequential correlations are transformed into temporal and spatial correlations. To deal with temporal and spatial correlations, we extend concepts which have successfully been used for transition probability analysis of combinational circuits. To cope with temporal correlations, Markov chains are applied. Markov chains have originally been proposed for estimation of combinational circuits in [14, 10, 13]. Spatial correlations of internal signals are captured based on reconvergence analysis. Reconvergence analysis has been used to analyse power in combinational circuits in [2, 13]. Our technique combines the concepts of unrolling, Markov Chains, and reconvergence regions such that for a sequential circuit a new system of equations results. The Picard-Peano method is applied similar to [17] to solve this system of equations.

The paper is organized as follows. Section 2 briefly reviews the concept of unrolling to transform sequential correlations into temporal and spatial correlations. Section 3 introduces suitable functions for signal and transition probability computation. In Section 4, we show how to cope with spatial correlations. Temporal correlations are regarded in Section 5. Section 6 is concerned with solving the obtained system of non-linear equations. In Section 7 detailed experimental results show the importance of accounting for spatial and temporal correlations and the high accuracy and efficiency of our new approach.

2. Unrolling Sequential Circuits

Sequential circuits can be cyclic as shown in Fig. 2. This cyclic structure introduces sequential correlations. To transform these sequential correlations into temporal and spatial correlations, we use the idea of unrolling similar to [6, 17].

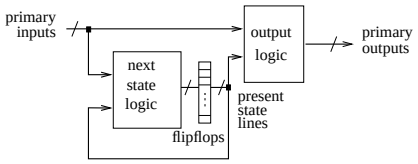

Figure 2. Cyclic structure of a sequential circuit

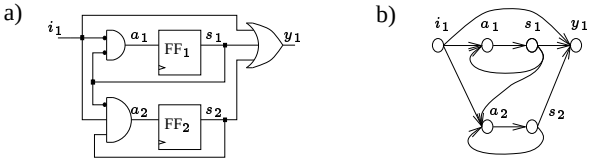

Figure 3. Example: a) circuit C1 and b) graph G1

The example of Fig. 3 is used for illustration. Circuit C1 in Fig. 3a) is represented by the graph G1 in Fig. 3b). For ease of notation, each node of G1 is labeled with the name of the output signal of the corresponding gate. A variable that corresponds to signal x_i at an arbitrary time t^0 is denoted by x_i^0. The variable x_i^T corresponds to signal x_i at time $t^0 - T$, where T is the duration of one clock cycle. Circuit C1 is determined by:

$$\begin{array}{lll} y_1^0 = i_1^0 + s_1^0 + s_2^0 & & s_1^0 = a_1^T \\ a_1^0 = \bar{i}_1^0 \bar{s}_1^0 & a_2^0 = i_1^0 \bar{s}_1^0 s_2^0 & s_2^0 = a_2^T \end{array} \quad (1)$$

To obtain an acyclic circuit, the next state logic is unrolled as shown in Fig. 4a). Unrolling the next state logic of circuit C1 yields circuit $C1^{ur}$ in Fig. 4b). Contrary to [6], each flipflop is substituted by a buffer. Note that the behavior of the circuit is preserved by substituting each variable x_i^0 in the transitive fanin of a flipflop by the corresponding variable of the previous clock cycle, x_i^T.

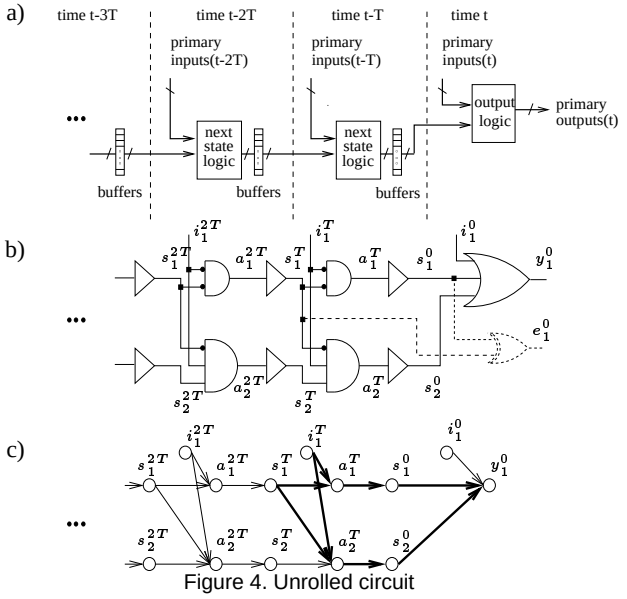

Figure 4. Unrolled circuit

3. Signal and Transition Probabilities

Our goal is to compute signal and transition probabilities. Let y correspond to an internal or an output signal. The Boolean function at y is denoted by $y = f(\underline{x})$, where $\underline{x} = (x_1, x_2, \ldots, x_n)$ is a vector of internal or primary input variables. How to find the most suitable function $f(\underline{x})$ will be discussed in the next section. The signal probability of y is computed on the function $f(\underline{x})$, i.e.:

$$p(y) = p(f(\underline{x})) = p(f) \quad (2)$$

If a transition on signal y occurs then y differs at two consecutive points in time, i.e., $y^T \oplus y^0$ is true. The transition probability is determined by:

$$E(y) = p\left(y^T \oplus y^0\right)$$

Thus, we think an XOR gate is inserted for each signal. For example for signal s_1 in Fig. 4b), an extra XOR-gate has been inserted to illustrate the transition probability computation of signal s_1. Thus we obtain

$$E(s_1) = p(s_1^T \oplus s_1^0) = p(e_1^0)$$

where $p(e_1^0)$ can be computed in the same fashion as Eqn. (2). This maps the problem of finding signal and transition probabilities to the problem of finding probabilities of the corresponding Boolean functions.

4. Spatial Correlations

In this section, we show how to find Boolean functions that are suitable to compute signal and transition probabilities. Such functions must have the following properties. Firstly, the variables of each function must correspond to spatially uncorrelated signals. This property is required by the technique given in Sections 5 and 6 to compute the probability of each function accurately. Secondly, the functions must be found such that the representation size of each function is as small as possible to achieve low CPU times. We use BDDs [1] for representation.

For each signal under estimation, we individually determine a circuit partition that has spatially uncorrelated partition inputs. The function and thus the BDD for this partition depends on spatially uncorrelated variables, only. Furthermore, the BDD size is correlated with the partition size and with the number of variables the BDD depends on. Therefore, we determine the partition with spatially uncorrelated inputs such that it has minimum number of internal signals and minimum number of inputs.

4.1. Reconvergence Regions

We now introduce maximal combined reconvergence regions and show that they are the partition that meet the above introduced requirements.

For ease of explanation, we use a Boolean network to give some basic definitions necessary to understand the analysis of reconvergence regions. The graph representation of an unrolled sequential circuit as shown in the example in Fig. 4c) becomes a Boolean network if the next state logic is unrolled only k instead of infinite times. For the definition of a Boolean network, please see [5]. Fig. 5a) gives an example.

The analysis of reconvergent fanouts originated in the area of fault simulation [9]. The key concepts introduced below have been derived from that work. We show how to use reconvergence analysis to obtain functions with mutually independent inputs.

Node p is a *primary reconvergent fanout stem* for the *reconvergence node* v, $PRFS(v)$, if two disjoint paths from p to v exist:

$$\exists_{P_i(p,v)} \quad \exists_{P_j(p,v) \neq P_i(p,v)} \left(set(P_i(p,v)) \cap set(P_j(p,v)) = \{p, v\} \right)$$

A subgraph of the Boolean network called *primary reconvergence region*, PRR_v, is the collection of all paths from all $\text{PRFS}(v)$ to v. A $\text{PRFS}(v)$ y is not included in PRR_v, unless there exists a $\text{PRFS}(v)$ x such that y is on a path from x to v. In graph G2 in Fig. 5a) nodes p and i are primary reconvergence fanout stems of v. Hence, the primary reconvergence region of v consists of nodes $\{o, p, t, u, v\}$. Note: i is not included in PRR_v.

The subgraph of a Boolean network that is constructed by the following rules is called a *maximal combined reconvergence region* of node v MCRR_v.

$$\begin{aligned} \text{MCRR}_v{}^{(0)} &= \text{PRR}_v \\ \text{MCRR}_v{}^{(n+1)} &= \text{MCRR}_v{}^{(n)} \cup \bigcup_{x \in \text{MCRR}_v{}^{(n)}} \text{PRR}_x \\ \text{MCRR}_v &= \text{MCRR}_v{}^{\infty} \end{aligned}$$

A $\text{PRFS}(x)$ is called *secondary reconvergent fanout stems* of node v, $\text{SRFS}(v)$, if x is included in MCRR_v.

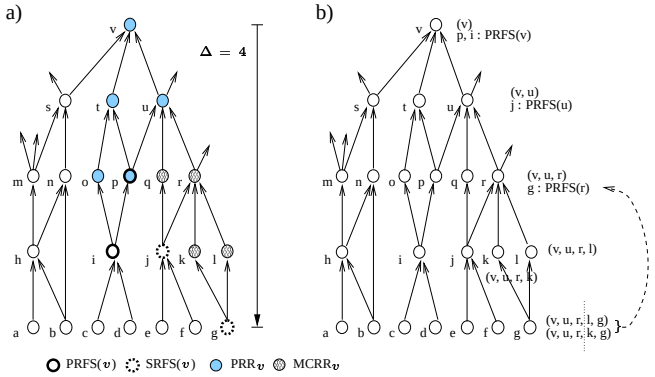

Figure 5. Graph G2

In graph G2 in Fig. 5a), node j is a $\text{SRFS}(v)$ and nodes q, r, u are included in $\text{MCRR}_v{}^{(1)}$ because u is in PRR_v. Node g is a $\text{SRFS}(v)$ and nodes k, l, r are included in $\text{MCRR}_v{}^{(2)}$ because r is in $\text{MCRR}_v{}^{(1)}$. After step two, all nodes of MCRR_v are determined. MCRR_v consists of nodes $k, l, o, p, q, r, t, u, v$.

Inputs to a maximal combined reconvergence region MCRR are the nodes that have no fanin and at least one fanout node in the MCRR. In graph G2 the inputs to MCRR_v are nodes $\{g, i, j, s\}$.

4.2. Exact Analysis

We look for internal signals that are spatially uncorrelated. If primary inputs are assumed to be spatially uncorrelated, spatial correlations between two internal signals are due to common ancestors. Since inputs to an MCRR cannot have any common ancestor, the presented technique guarantees spatially uncorrelated inputs to an MCRR.

For every partition with fewer nodes than the MCRR, some inputs have a common ancestor. These inputs are spatially correlated. We can therefore conclude that the MCRR of a signal is the smallest subgraph with spatially uncorrelated inputs.

Furthermore, it can be shown that no subgraph with spatially uncorrelated but fewer inputs than the MCRR exists. We will give an idea of how to prove this. A subgraph with spatially uncorrelated inputs must include the MCRR. Therefore, each input to the subgraph must be an input to the MCRR or an ancestor of an input to the MCRR. Thus, if such a subgraph with less inputs than the MCRR exists, then at least two inputs of the MCRR have a common ancestor, which contradicts the definition of the MCRR.

4.3. Approximation

For many nodes, no reconvergent fanout stem exists and thus the MCRR consists of the reconvergence node, only. For some nodes, however, the MCRR covers almost all ancestor nodes, which can be an infinite number of nodes for an unrolled sequential circuit. We therefore suggest an approximation technique to trade off estimation accuracy for CPU time and memory resources.

For a computationally inexpensive approximation we introduce a parameter Δ. Then, we search for those reconvergent fanout stems, for which the number of nodes on every path from the fanout stem to the reconvergence node v is less or equal than Δ. The subgraph induced by these paths is called *combined reconvergence region* $\text{CRR}_{\Delta,v}$. We have to tolerate that some reconvergence fanout stems have possibly not been found and therefore some inputs to a $\text{CRR}_{\Delta,v}$ might be spatially correlated. Our experimental results demonstrate that this approximation causes only small inaccuracies.

Output y of Circuit C1 is used for illustration. For $\Delta = 4$, we determine $\text{CRR}_{4,y}$, which is the subgraph marked by bold lines in Fig. 4c). Thus, we obtain

$$p(y) = p(f(i_0^0, i_0^T, s_1^T, s_2^T))$$

For $\Delta = 6$, we find $\text{CRR}_{6,y}$, which yields

$$p(y) = p(f(i_0^0, i_0^T, i_0^{2T}, s_1^{2T}, s_2^{2T}))$$

The parameter Δ is a user input and allows a trade-off between estimation accuracy and CPU resources. Increasing the value of Δ makes the estimation more accurate, but BDD sizes and thus CPU times rise. This trade-off is evaluated in the result section.

4.4. Implementation

First, all PRFS and SRFS for a node b are computed. As soon as they are detected, the $\text{CRR}_{\Delta,b}$ and its inputs can easily be determined. To find PRFS and SRFS, the implementation technique suggested in [9] could be used if all edges of our network are inverted. This implementation is efficient, only if PRFS and SRFS of all nodes must be determined. Since logic synthesis changes small partitions we have to determine the CRR for few or only one node. Therefore, we propose a new implementation technique.

Our implementation technique detects all primary and secondary reconvergence fanout stems in the fanin region of one node v. For this purpose, the transitive fanin is traversed backwards from the reconvergence node v with depth-first search. At each visited node g, the path from v to g is assigned. If a node g has already assigned a path, g is a PRFS to some reconvergence node. The reconvergence node is determined by comparing the path already assigned to node g with the presently discovered path. The PRFS is assigned to its reconvergence node. $\text{CRR}_{\Delta,v}$ can be constructed by starting at node v and recursively collecting all nodes and their PRFSs on all paths from any collected PRFS to the reconvergence node.

The traversal and the assigned paths up to node g are illustrated in Fig. 5b). When node g is visited for the second time, the two paths at node g are compared. The comparison shows that node r is the last common node of both paths. Thus, node g is determined to be a PRFS of node r.

5. Temporal Correlations

To compute probabilities, we need mutually statistically independent variables, i.e., spatially and temporally uncorrelated variables. Reconvergence analysis was employed to obtain functions with spatially uncorrelated variables. But some variables may be temporally correlated. Therefore, we decompose the functions

in mutually uncorrelated subfunctions. To achieve this, variables are divided into packets. Each packet contains all variables that correspond to one signal at different points in time. Thus, each packet contains temporally correlated variables but each variable of a packet is uncorrelated to any variable of another packet. Recursively, each packet is decomposed from the considered function. Finally, the decomposed subfunctions are small and their probability can efficiently be computed with a technique based on the concept of Markov Chains.

5.1. Decomposition with Respect to Temporal Correlations

Let us consider a function f, which depends on the variables $x_i^{kT}, \ldots, x_i^0$ and on other variables. Each variable x_i^{vT} corresponds to signal x_i at time $t_0 - vT$. Function f is independent of x_i^{vT} for $v > k$. With $m_j(x_i)$ we denote the conjunction of the variables $x_i^{kT}, \ldots, x_i^0$ where the binary representation of value j determines the phase of each variable: $m_0(x_i) = x_i^{kT} \ldots x_i^T x_i^0$; $m_1(x_i) = x_i^{kT} \ldots x_i^T \overline{x}_i^0$; ...; $m_{2^k-1}(x_i) = \overline{x}_i^{kT} \ldots \overline{x}_i^T \overline{x}_i^0$. The probability of $x_i = 0$ is denoted by $p(\overline{x}_i)$ and evaluates to $p(\overline{x}_i) = 1 - p(x_i)$.

Let $f|_{m_j(x_i)}$ be the cofactor of f with respect to $m_j(x_i)$. Applying the Shannon decomposition ($f = x_i f|_{x_i} + \overline{x}_i f|_{\overline{x}_i}$) on the transition function several times yields:

$$f = \sum_{j=0}^{2^k-1} m_j(x_i) f|_{m_j(x_i)} \tag{3}$$

Since $m_j(x_i)$ contains all variables that correspond to x_i, each cofactored function $f|_{m_j(x_i)}$ is independent of x_i.

5.2. Probability of a Boolean Function

To compute the probability of a Boolean function, we start with Eqn. (3). The probability $p(f)$ is thus given by:

$$p(f) = p\Big(\sum_{j=0}^{2^k-1} m_j(x_i) f|_{m_j(x_i)}\Big)$$

For each assignment of values to $x_i^{kT}, \ldots, x_i^0$, exactly one $m_j(x_i)$ evaluates to one. Thus, all terms $m_j(x_i)$ are mutually disjoint and we can write:

$$p(f) = \sum_{j=0}^{2^k-1} p(m_j(x_i) f|_{m_j(x_i)})$$

As already mentioned, each $f|_{m_j(x_i)}$ is independent of x_i. Therefore, $m_j(x_i)$ and $f|_{m_j(x_i)}$ are mutually independent and we obtain:

$$p(f) = \sum_{j=0}^{2^k-1} p(m_j(x_i))\, p(f|_{m_j(x_i)}) \tag{4}$$

Such a decomposition of function f is performed for each variable x_i. Decomposing the last variable x_n yields either 1 or 0. Thus, we obtain a sum of products of probabilities $p(m_j(x_i))$. The remaining problem is to compute the probabilities $p(m_j(x_i))$.

Computing Eqn. 4 can efficiently be performed on BDDs. A description of the implementation if only two points in time are considered is given in [13]. We extended this technique to k points in time. The computation complexity for evaluating the transition probability is linear in terms of BDD nodes.

5.3. Probability of Conjunctions of Temporally Correlated Variables

We consider each primary input and each internal signal as a discrete, stationary, first order Markov process. Then, the computation of the probabilities $p(m_j(x_i))$ can be derived from a Markov chain. A Markov chain for x_i is shown in Fig. 6.

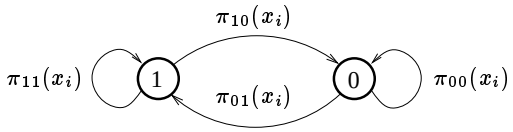

Figure 6. Markov chain

The probability $\pi_{10}(x_i)$ in Fig. 6 is defined by $\pi_{10}(x_i) = p(x_i^T \overline{x}_i^0 | x_i^T)$, i.e., $\pi_{10}(x_i)$ is the probability that a $1 \to 0$ transition on x_i occurs from $t^0 - T$ to t^0, if x_i was 1 at $t^0 - T$. Let $\tilde{x}_i$ be either x_i or $\overline{x}_i$. Furthermore, $\sim$ is 1 or 0 accordingly, so that we can write $\pi_{\sim\sim} = p(\tilde{x}_i^T \tilde{x}_i^0 | \tilde{x}_i^T)$. The literature, e.g., [12], provides useful equations:

$$p(\tilde{x}_i^T \tilde{x}_i^0) = p(\tilde{x}_i^T) p(\tilde{x}_i^T \tilde{x}_i^0 | \tilde{x}_i^T) = p(\tilde{x}_i^T) \pi_{\sim\sim}(x_i)$$

Since the signals' behavior is assumed to be Markovian, the probabilities $p(m_j(x_i))$ can be described by the product of probabilities $\pi_{\sim\sim}(x_i)$. This is shown for the example $j = 6$ and $k = 2$:

$$\begin{aligned} p(m_6(x_i)) &= p(\overline{x}_i^{2T} \overline{x}_i^T x_i^0) \\ &= p(\overline{x}_i^{2T}) \cdot p(\overline{x}_i^{2T} \overline{x}_i^T | \overline{x}_i^{2T}) \cdot p(\overline{x}_i^T x_i^0 | \overline{x}_i^T) \\ &= p(\overline{x}_i) \cdot \pi_{00}(x_i) \cdot \pi_{01}(x_i) \end{aligned}$$

From [12], several equations can be obtained for a Markov chain:

$$\begin{aligned} 1 &= p(x_i) + p(\overline{x}_i) \\ 1 &= \pi_{11}(x_i) + \pi_{10}(x_i) \\ 1 &= \pi_{01}(x_i) + \pi_{00}(x_i) \\ p(x_i) &= p(x_i)\pi_{11}(x_i) + p(\overline{x}_i)\pi_{01}(x_i) \\ p(\overline{x}_i) &= p(\overline{x}_i)\pi_{00}(x_i) + p(x_i)\pi_{10}(x_i) \end{aligned} \tag{5}$$

A transition occurs on x_i if x_i changes its value either from 1 to 0 or from 0 to 1.

$$E(x_i) = p(x_i^T \overline{x}_i^0) + p(\overline{x}_i^T x_i^0) \tag{6}$$

With Eqns. (5) and (6), the probabilities $\pi_{\sim\sim}(x_i)$ can be expressed by $p(x_i)$ and $E(x_i)$:

$$\begin{aligned} \pi_{11}(x_i) &= 1 - \tfrac{E(x_i)}{2p(x_i)} & \pi_{10}(x_i) &= \tfrac{E(x_i)}{2p(x_i)} \\ \pi_{00}(x_i) &= 1 - \tfrac{E(x_i)}{2p(\overline{x}_i)} & \pi_{01}(x_i) &= \tfrac{E(x_i)}{2p(\overline{x}_i)} \end{aligned} \tag{7}$$

Obviously, Eqn. (7) are not valid if $p(x_i) = 0$ or if $p(\overline{x}_i) = 0$. If x_i always is 0 ($p(x_i) = 0$) or if x_i always is 1 ($p(\overline{x}_i) = 0$) then no transition will occur at all ($E(x_i) = 0$) and therefore, $\pi_{11}(x_i) = \pi_{00}(x_i) = 1$ and $\pi_{10}(x_i) = \pi_{01} = 0$ hold.

6. System of Equations

The combination of the techniques presented in Sections 2 to 5 yields for each internal signal an equation for the signal probability and an equation for the transition probability. These equations depend only on signal probabilities and transition probabilities of internal signals and primary inputs. Generally, for a circuit with n internal signals $x_1, \ldots, x_n$ and m primary inputs $i_1, \ldots, i_m$, we obtain:

$$\begin{aligned} p(x_1) &= f_1(p(x_1), \ldots, p(x_n), p(i_1), \ldots, p(i_m), \\ &\quad\ \vdots \qquad E(x_1), \ldots, E(x_n), E(i_1), \ldots, E(i_m)) \\ p(x_n) &= f_n(\ \ldots\ ) \\ E(x_1) &= f_{n+1}(\ \ldots\ ) \\ &\quad\ \vdots \\ E(x_n) &= f_{2n}(\ \ldots\ ) \end{aligned} \tag{8}$$

These equations account for sequential, temporal, and spatial correlations. Furthermore, according to section 4, small equations are obtained. The 2n equations build a set of non-linear equations. Solving this system of non-linear equations yields the required signal and transition probabilities. As suggested in [17], we use the Picard-Peano technique to solve the system of equations.

Regarding the system of equations, there are two differences to [17]. Firstly, we have equations for both signal and transition

probabilities. Therefore, two equations per signal appear. Secondly, in [17], each function at a state line is computed in terms of primary inputs and state lines. We obtain equations not only for state lines but also for signals internal to the next state logic. Therefore, the system of equation contains a larger number of equations. However, these equations are significantly smaller and thus the evaluation of such an equation is computationally less expensive. The overall computation complexity decreases, allowing to quickly evaluate large sequential circuits. While for circuit `s13207`, 338 minutes are required in [17] to estimate signal probabilities of state lines on SUN Sparc 2, our approach computes signal and transition probabilities of all signals in less than 211 seconds on DEC 3000 Model AXP.

7. Results

Results are computed for a large set of sequential ISCAS and LGSynth91 benchmark circuits. We mapped them to the `mcnc.genlib` library with the SIS [16] technology mapper after having applied the SIS script `script.rugged`.

7.1. Impact of Temporal Correlations in Sequential Circuits

Neglecting temporal correlations at primary inputs causes inaccurate values for signal and transition probabilities at internal signals. To demonstrate this, we performed four simulations for each circuit with 10^5 input vectors. Every input was assigned signal probability $p = 0.5$ for all four simulations. The transition probability of every primary input was set to 0.5 for the first simulation, to 0.25 for the second, to 0.10 for the third, and to 0.02 for the fourth. For pseudo-randomly created input vectors, $E = 0.5$ at primary inputs yields temporally uncorrelated primary input vectors. Thus, the first simulation determines transition probabilities at internal signals with temporally uncorrelated primary inputs. With decreasing transition probabilities at primary inputs, the temporal correlations of primary inputs increase. Techniques that neglect temporal correlations at primary inputs produce the same values for the transition probability of an internal signal regardless of the actual transition probabilities at primary inputs, i.e., these techniques yield the result of the first simulation for any assignment of E to primary inputs. Therefore, the simulations with $E \neq 0.5$ at primary inputs are compared to the simulation with $E = 0.5$ at primary inputs. In Tab. 1, the obtained average absolute error is shown. The average absolute error is the sum of the absolute value of the error for each signal divided by the number of signals. As Tab. 1 shows, the average absolute error increases up to more than 0.21 for circuit `s1196`.

circuit	$E = 0.25$	$E = 0.10$	$E = 0.02$
s208	0.080	0.129	0.156
s298	0.059	0.101	0.125
s382	0.042	0.068	0.082
s444	0.041	0.066	0.080
s526	0.037	0.059	0.073
s820	0.073	0.126	0.161
s1196	0.083	0.160	0.216
s1423	0.065	0.112	0.143
s5378	0.052	0.100	0.145
s9234	0.030	0.049	0.063
s13207	0.036	0.058	0.071
mm30	0.075	0.140	0.183
dsip	0.080	0.153	0.203

Table 1. Average absolute error for neglecting temporal correlations at primary inputs.

7.2. Our Analysis Technique

To show the accuracy of the presented analysis technique, our results are compared to a simulation with 10^5 patterns. The average absolute error and the CPU times in seconds are reported in Tables 2 and 3, respectively, for various values of Δ. First, for each input, $p = 0.5$ and $E = 0.25$ was selected.

Increasing inaccuracies for decreasing Δ indicate that spatial correlations must not be neglected. A comparison of the error due to neglecting temporal correlations at primary inputs (column "$E = 0.25$" in Tab. 1) with the error of our technique (column "$\Delta = 10$" in Tab. 2) shows that for all circuits with exception of `s9234`, our technique causes the smaller inaccuracies. This demonstrates that no technique that neglects temporal correlations at primary inputs can be as accurate as our technique. Of course, the error of an approximation technique that neglects temporal correlations at primary inputs will be even larger than the error of column "$E = 0.25$" in Tab. 1, since approximations cause additional inaccuracies. Furthermore, the error due to neglecting temporal correlation at primary inputs increases with increasing temporal correlations as Tab. 1 indicates. Our technique, however, is almost insensitive to increasing temporal correlations at primary inputs. The last column in Tab. 2 shows the average absolute error for strongly temporally correlated primary inputs (E = 0.02). This error is slightly higher than the error for E = 0.25, but it is by far smaller than the error in the last column in Tab. 1. When neglecting temporal correlations at primary inputs for circuit `s1196`, e.g., the average absolute error is 0.216 whereas our technique yields an error of only 0.007.

circuit	$E = 0.25$			$E = 0.02$
	$\Delta = 0$	$\Delta = 5$	$\Delta = 10$	$\Delta = 10$
s208	0.210	0.021	0.005	0.008
s298	0.064	0.061	0.058	0.090
s382	0.039	0.031	0.018	0.040
s444	0.074	0.031	0.024	0.049
s526	0.023	0.021	0.022	0.042
s820	0.022	0.019	0.022	0.033
s1196	0.017	0.005	0.004	0.007
s1423	0.054	0.047	0.032	0.070
s5378	0.016	0.011	0.007	0.018
s9234	0.109	0.074	0.067	0.055
s13207	0.128	0.050	0.030	0.039
mm30	0.052	0.034	0.026	0.045
dsip	0.040	0.010	0.017	0.019

Table 2. Average absolute error for our technique.

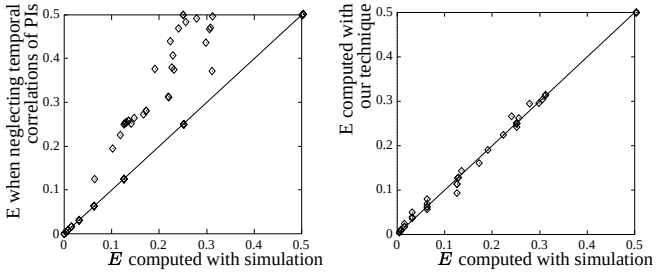

Figure 7. Signal probabilities of the signals of circuit s208

The scattered diagrams in Fig. 7 illustrate the results for circuit `s208`. For primary inputs, we assume $p = 0.5$ and $E = 0.25$. In both diagrams, the horizontal axis gives the correct transition probabilities computed with simulation. The vertical axis of the left diagram shows transition probabilities when temporal correlations are neglected. In the right diagram, transition probabilities are computed by our technique. While the results of our technique closely meet the correct transition probabilities, the results under neglect of transition probabilities at primary inputs are far off.

Column "# FFs" in Tab. 3 gives the number of flipflops for each circuit. Also in Tab. 3, CPU times in seconds on DEC 3000 Model AXP workstations are presented for $\Delta = 0$, $\Delta = 5$, and $\Delta = 10$. Comparing the accuracy results in Tab. 2 with the CPU times in Tab. 3 shows that various assignments to the user parameter Δ allow different trade-offs between accuracy and computation costs.

8. Conclusion

Transition probability analysis for sequential circuits is a difficult task but crucial for low power optimization at logic level. So far, probabilistic approaches neglected temporal correlations at primary inputs or could not handle large circuits. We have presented a novel approach to transition probability analysis, which accounts for temporal correlations at primary inputs as well as for sequen-

circuit	# FFs	$\Delta = 0$	$\Delta = 5$	$\Delta = 10$
s208	8	0.6	0.3	1.2
s298	14	0.3	0.9	3.1
s382	21	0.8	1.3	4.9
s444	21	0.5	1.5	5.2
s526	21	0.6	1.5	5.8
s820	5	0.6	5.9	17.5
s1196	18	0.9	4.6	13.7
s1423	74	2.5	6.4	30.8
s5378	162	12.7	129.6	284.2
s9234	135	5.7	231.7	318.6
s13207	474	14.6	88.9	210.5
mm30	90	2.9	69.1	240.8
dsip	224	12.0	37.3	141.2

Table 3. Number of flipflops and CPU times in seconds.

tial correlations and spatial correlations of internal signals. This is achieved by exploiting and combining the concepts of unrolling, reconvergence analysis, decomposing packets of temporally correlated variables, and Markov chains. Experimental results demonstrate the accuracy and efficiency of our novel approach even for large circuits with several hundreds of flipflops.

References

[1] R. E. Bryant. Graph–based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C–35(8):677–691, Aug. 1986.

[2] D. I. Cheng, M. Marek-Sadowska, and K.-T. Cheng. Speeding Up Power Estimation by Topological Analysis. In *IEEE Custom Integrated Circuits Conference (CICC)*, May 1995.

[3] T.-L. Chou and K. Roy. Estimation of sequential ciruit activity considering spatial and temporal correlations. In *IEEE International Conference on Computer Design (ICCD)*, pages 577–582, Oct. 1995.

[4] T.-L. Chou and K. Roy. Statistical Estimation of Sequential Circuit Activity. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 34 – 37, Nov. 1995.

[5] G. de Micheli. *Synthesis and Optimization of Digital Circuits*. McGraw–Hill, Inc., New York, 1994.

[6] S. Dey, F. Brglez, and G. Kedem. Partitioning Sequential Circuits for Logic Optimization. In *IEEE International Conference on Computer Design (ICCD)*, pages 70 – 76, Oct. 1991.

[7] L. Glasser and D. Dobberpuhl. *The Design and Analysis of VLSI Circuits*. Addison-Wesley Publishing Company, 1985.

[8] G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Probabilistic Analysis of Large Finite State Machines. In *ACM/IEEE Design Automation Conference (DAC)*, pages 270 – 275, June 1994.

[9] F. Maamari and J. Rajski. A reconvergent fanout analysis for efficient exact fault simulation of combinational circuits. In *IEEE International Symposium on Fault-Tolerant Computing (FTCS)*, pages 122–127, June 1988. volume = 18.

[10] R. Marculescu, D. Marculescu, and M. Pedram. Efficient Power Estimation for Highly Correlated Input Streams. In *ACM/IEEE Design Automation Conference (DAC)*, June 1995.

[11] F. N. Najm, S. Goel, and I. N. Hajj. Power Estimation in Sequential Circuits. In *ACM/IEEE Design Automation Conference (DAC)*, June 1995.

[12] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1991.

[13] P. Schneider, U. Schlichtmann, and B. Wurth. Fast power estimation of large circuits. *IEEE Design & Test of Computers*, pages 70 – 78, Spring 1996.

[14] P. H. Schneider and U. Schlichtmann. Decomposition of Boolean Function for Low Power Based on a New Power Estimation Technique. In *IEEE International Workshop on Low Power Design IWLPD*, pages 123 – 128, Apr. 1994.

[15] P. H. Schneider and B. Wurth. Transition probability estimation for combinational and sequential circuits. In *International Workshop on Logic Synthesis (IWLS)*, pages P4b1 – P4b10, May 1995.

[16] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephen, R. K. Brayton, and A. Sangiovanni-Vincentelli. SIS: A system for sequential circuit synthesis. Technical report, Electronics Research Laboratory, Memorandum No. UCB/ERL M92/41, May 1992.

[17] C.-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain, and B. Lin. Power estimation methods for sequential logic circuits. *IEEE Transactions on VLSI*, pages 404 – 416, Sept. 1995.

Appendix

Neglecting temporal correlations at primary inputs causes major inaccuracies. This does not only hold for transition probability analysis but also for state probability computation. We demonstrate this for the example in Fig. 8a). To compute the state probabilities, we employ the technique based on Chapman-Kolmogorov equations. This technique is known to be exact if primary inputs are uncorrelated.

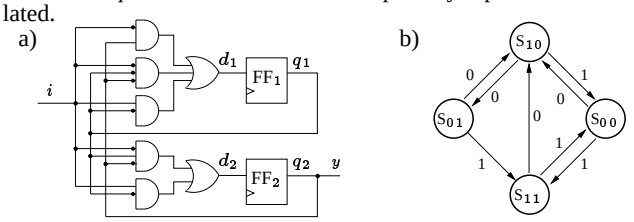

Figure 8. Sequential circuit and state transition graph

Fig. 8a) shows a sequential circuit. The Boolean functions of the d-inputs of the two flipflops are given as follows:

$$d_1 = \overline{i}\, q_2 + \overline{i}\, \overline{q}_1\overline{q}_2 + i\, \overline{q}_1$$
$$d_2 = \overline{i}\, q_1\overline{q}_2 + i\, \overline{q}_1$$

We assume the four states to be encoded as follows:

$$\mathrm{S}_{00} : \overline{q}_1\overline{q}_2 \qquad \mathrm{S}_{01} : \overline{q}_1 q_2$$
$$\mathrm{S}_{10} : q_1\overline{q}_2 \qquad \mathrm{S}_{11} : q_1 q_2$$

From the according state transition graph shown in Fig. 8b), the Chapman-Kolmogorov equations can easily be derived:

$$\begin{aligned}
p(\mathrm{S}_{00}) &= p(i)p(\mathrm{S}_{10}) + p(i)p(\mathrm{S}_{11}) \\
p(\mathrm{S}_{01}) &= p(\overline{i})p(\mathrm{S}_{10}) \\
p(\mathrm{S}_{10}) &= p(\overline{i})p(\mathrm{S}_{00}) + p(\overline{i})p(\mathrm{S}_{01}) + p(\overline{i})p(\mathrm{S}_{11}) \\
p(\mathrm{S}_{11}) &= p(i)p(\mathrm{S}_{00}) + p(i)p(\mathrm{S}_{01}) \\
1 &= p(\mathrm{S}_{00}) + p(\mathrm{S}_{01}) + p(\mathrm{S}_{10}) + p(\mathrm{S}_{11})
\end{aligned}$$

Let us assume the signal probability at the input signal i to be one half, i.e., $p(i) = p(\overline{i}) = 0.5$. Solving the linear system of equations yields the following state probabilities:

$$\begin{aligned}
p(\mathrm{S}_{00}) &= 0.28 \qquad p(\mathrm{S}_{01}) = 0.17 \\
p(\mathrm{S}_{10}) &= 0.33 \qquad p(\mathrm{S}_{11}) = 0.22
\end{aligned} \tag{9}$$

This result is correct if signal i is temporally uncorrelated. Let us now consider the state probabilities in case of a temporally correlated input i. Transition probability $E(i) = 1$ is used for illustration. The signal probability of input i is set to 0.5 again. Thus, techniques that neglect temporal correlations at primary inputs still yield the result of Eqns. (9). For $E(i) = 1$, signal i makes a transition every clock cycle. Thus, signal i takes on values "… 0101010101 …". The state transition graph in Fig. 8b) shows that after at most 3 clock cycles, the present state is alternately S_{00} and S_{10}. Thus, for $p(i) = 0.5$ and $E(i) = 1$, the state probabilities are:

$$\begin{aligned}
p(\mathrm{S}_{00}) &= 0.5 \qquad p(\mathrm{S}_{01}) = 0.0 \\
p(\mathrm{S}_{10}) &= 0.5 \qquad p(\mathrm{S}_{11}) = 0.0
\end{aligned}$$

Let us assume $E(i) = 0.0001$ now. Thus, on average the signal i is "1" in 10,000 consecutive clock cycles and then "0" in 10,000 consecutive clock cycles and so on, i.e., signal i takes on values "111…111000…000111…". The state transition graph in Fig. 8b) shows that on average, the present state alternates 10,000 times between S_{10} and S_{01} and then 10,000 times between S_{00} and S_{11}. Thus, for $p(i) = 0.5$ and $E(i) = 0.0001$, the state probabilities are:

$$\begin{aligned}
p(\mathrm{S}_{00}) &= 0.25 \qquad p(\mathrm{S}_{01}) = 0.25 \\
p(\mathrm{S}_{10}) &= 0.25 \qquad p(\mathrm{S}_{11}) = 0.25
\end{aligned}$$

Obviously, state probabilities in Eqns. (9) differ from the results with temporally correlated primary input i. Neglecting transition probabilities of primary inputs causes major inaccuracies in computing state probabilities. Deriving state line probabilities from these state probabilities propagates the inaccuracy. Also, computing transition probabilities based on inaccurate state line probabilities yields inaccurate values.