A Fast and Accurate Method of Power Estimation for Logic Level Networks*

G. THEODORIDIS*, S. THEOHARIS*, D. SOUDRISb† and C. GOUTIS*

*VLSI Design Lab, Dept. of Electrical and Computer Eng., University of Patras, Patras 26110, Greece;
†VLSI Design and Testing Center, Dept. of Electrical and Computer Eng.,
Democritus Univ. of Thrace, 67100 Xanthi, Greece

(Received 20 June 2000; In final form 3 August 2000)

A method for estimating the power consumption of multilevel combinational networks is introduced. The proposed method has as inputs the signal probabilities, the data correlations of the primary inputs and the structure of the circuit, and consists of two major steps: (i) the calculation of the switching activity on an individual gate and (ii) the calculation of the switching activity of any node of the network. The foregoing step includes the derivation of novel formulas for calculating the switching activity of basic gates. The latter step includes the development of an algorithm, which propagates the signal probabilities through the network and calculates the switching activity of any logic node. The proposed method provides accurate switching activity values performing their calculation in reduced time interval. The experimental results prove that the proposed method achieves significant reduction up to 50% in terms of multiplications compared to method of [6].

Keywords: Low power design; Switching activity estimation; Power dissipation model; Markov chains; Temporal and spatial correlation; CMOS combinational circuits

1. INTRODUCTION

The requirement for long battery life at the wide spread portable communication systems forces the designers to take into consideration except of the two traditional parameters, area and speed, and a third one, the power consumption [1, 2].

Recently, many researchers have suggested a number of methods for estimating the power consumption in digital CMOS circuits [3–7]. In particular, using symbolic simulation and Ordered Binary Decision Diagrams (OBDDs) [10, 11], and considering spurious transitions, temporal correlation, and structural correlation a power estimator was introduced by Devadas et al. [5]. The main drawback of the method is its high computational complexity. Also, the spatial correlations of the input signals are not included. Schneider et al. [7]

---

* This work was supported by the project LPGD 25256 ESPRIT IV of European Union and Intracom S.A.
† Corresponding author. Tel.: +30 541 79557, e-mail: dsoudris@ee.duth.gr
presented a method using Markov chain theory, Reduced OBDDs, first-order temporal correlation, and structural correlation. However, it has been assumed that the primary inputs are independent and spatially-uncorrelated. Marculescu et al. [6] based on [9], on OBDDs and on Markov chain theory, suggested an accurate model for power estimation, using probabilistic methods. Introducing the concept of the transition correlation coefficient (TCC), the spatiotemporal correlation is taken into account. Two methods, namely the global and incremental method, have been proposed. Comparing these methods, it is concluded that the latter method is less accurate than the first one. The common drawback is the large computational complexity.

In this paper a novel method for estimating the switching activity of a multilevel combinational circuit, is introduced. The proposed method belongs to the class of probabilistic approaches [12] and provides highly accurate switching activity values in smaller time interval than the existing methods. The method has as inputs the static probability, the transition probability, and the data correlations of the primary inputs, as well as, the after mapping structure of a logic network. The network consists of zero-delay basic logic gates, e.g., AND, OR, and NOT. The outcome of the method is the switching activity value of any node of the logic network.

The proposed method is accurate and fast. More specifically, the accuracy arises from the fact that spatial, temporal, and structural correlations of the signals, at all logic levels are taken into account. The reduced time cost results from the reduced computational complexity in terms of multiplications. The proposed method comprises two steps: (i) switching activity estimation of a single basic gate and (ii) switching activity estimation of any node of the considered logic network. In particular, the switching activity of a single basic gate can be calculated by a set of newly-developed formulas. Detailed formal description of the derivation process of these formulas is given. The second step includes the development of a new procedure, which determines the propagation of the signal probabilities through the basic gates and the wires of the logic network, calculates the appropriate TCCs, and eventually, estimates the switching activity of a logic node. The computational complexity of the switching activity of a basic gate and logic network in terms of multiplications, have been computed by a series of formally proven lemmas. Moreover, the accuracy and complexity of the proposed method are compared with the ones of [6], using a number of lemmas. It is concluded that both methods have identical accuracy, while the proposed one has less complexity than [6]. The experimental results for a set of ISCAS circuits indicate remarkably reduced computational complexity.

The rest of the paper is organised as follows. Section 2, the proposed method for power estimation is presented in detail manner. The experimental results are shown in Section 3. Finally, the conclusions are given in Section 4.

2. THE PROPOSED METHOD

The estimation of the power consumption of a logic circuit can be stated as follows:

Assuming a without loops combinational logic network of n-input zero-delayed basic gates (i.e., AND, OR, NOT, etc.) and given the statistical properties of the primary inputs (transition probabilities and pairwise TCCs) estimate its switching activity and eventually, the power consumption.

The total number of switches of a circuit node is the sum of the functional transitions and the spurious transitions (glitches). The functional transitions come from the structure of the circuit and the applied input vectors, while the glitches arise mainly from the gate and wires delays.

In this paper only the functional transitions are considered since a zero delay model is assumed. Including the evaluation of the glitches the estimation becomes more accurate but the computational complexity is increased since additional parameters such as the circuit delay paths, the actual values of the capacitances of the nodes, the slope of the input pulses and the width of the input
pulses etc., have to be considered. However, in a synthesis environment, where a lot of power estimations are performed to characterize in terms of switching activity alternatives implementations of the circuit, a fast and as possible as accurate estimation is required. Adopting a method which considers the contribution of glitches the evaluated power becomes more accurate but the computational complexity is increased prohibitively. Moreover, some of the parameters such as the delay of the circuit lines and the capacitance of the nodes are not available in a gate level description. Thus, a zero gate delay model is used adopting the introduced error since the contribution of the glitches is ignored.

2.1. Switching Activity Estimation of a Single Gate

The formula, which calculates the switching activity of the output of a gate in terms of its inputs, will be derived first. To compute the switching activity of an n-input gate, we must consider the Boolean behaviour of a gate, which is specified by its controlling value. The output of an n-input basic gate performs a transition, when one or more inputs perform identical transitions simultaneously in two successive clock cycles and \( t + T \), while the remaining inputs are non-controlling value. Firstly, the switching activity of the output of a 3-input AND gate will be studied in detail manner. Then, the study will be generalised for an n-input AND gate. It must be noted that the same analysis holds for all basic gates.

2.1.1. Computation of the Switching Activity of a 3-input AND Gate

A 3-input AND gate, \( x_0, x_1, \ldots, x_{n-1} \), performs a \( TC^{x_0,x_1,x_2}_{11,01} \) transition in three cases:

(i) one input performs a \( 0 \rightarrow 1 \) (\( 1 \rightarrow 0 \)) transition simultaneously, while the remaining inputs are in high state, and

(ii) two inputs perform a \( 0 \rightarrow 1 \) (\( 1 \rightarrow 0 \)) transition simultaneously, while the remaining input is in high state, and

(iii) all inputs perform a \( 0 \rightarrow 1 \) (\( 1 \rightarrow 0 \)) transition simultaneously.

Consequently, the transition probability, \( p(y_{0 \rightarrow 1}) \), can be approximated by:

\[
p(y_{0 \rightarrow 1}) = p_{0 \rightarrow 1}^{x_0} p_{1 \rightarrow 0}^{x_1} p_{1 \rightarrow 0}^{x_2} + p_{0 \rightarrow 1}^{x_0} p_{1 \rightarrow 0}^{x_1} p_{1 \rightarrow 0}^{x_2} + p_{0 \rightarrow 1}^{x_0} p_{1 \rightarrow 0}^{x_1} p_{1 \rightarrow 0}^{x_2}
\]

\[
+ p_{0 \rightarrow 1}^{x_0} p_{1 \rightarrow 0}^{x_1} p_{0 \rightarrow 1}^{x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{1 \rightarrow 0}^{x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{1 \rightarrow 0}^{x_2}
\]

\[
+ p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2}
\]

\[
(1)
\]

The transition probability, \( p(x_{i \rightarrow j}) \), \( i, j \in \{0, 1\} \), corresponds to probability a signal \( x \) to perform a transition from state \( i \) to state \( j \) in two successive time clocks. Thus, the switching activity is evaluated considering the first order temporal correlation. The higher order temporal correlation should be considered in a similar way but the computational complexity while the accuracy is not improved significantly.

Furthermore, Eq. (1) is an approximation of the switching activity, since the data correlations of the input signals are not taken into account. Considering that the data correlations can be described by the appropriate TCCs, \( TC_{a,b,c}^{x_0,x_1,x_2} \), where \( a, b, c \in \{00, 01, 10, 11\} \), the appropriate TCCs must appear in each product of Eq. (1). Employing the formula \( TC_{a,b,c}^{x_0,x_1,x_2} = TC_{a,b}^{x_0,x_1} + TC_{b,c}^{x_0,x_1} + TC_{c,a}^{x_0,x_1} + TC_{a,b,c}^{x_0,x_1,x_2} \) \{6, 9\}, the first term of Eq. (1) can be expressed as:

\[
p_{0 \rightarrow 1}^{x_0} p_{1 \rightarrow 0}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2}
\]

\[
= p_{0 \rightarrow 1}^{x_0} p_{1 \rightarrow 0}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{1 \rightarrow 0}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{1 \rightarrow 0}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{1 \rightarrow 0}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2}
\]

\[
(2)
\]

Therefore,

\[
p(y_{0 \rightarrow 1}) = p_{0 \rightarrow 1}^{x_0} p_{1 \rightarrow 0}^{x_1} p_{1 \rightarrow 0}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{1 \rightarrow 0}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{1 \rightarrow 0}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{1 \rightarrow 0}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{1 \rightarrow 0}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{1 \rightarrow 0}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2}
\]

\[
+ p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2} + p_{0 \rightarrow 1}^{x_0} p_{0 \rightarrow 1}^{x_1} p_{0 \rightarrow 1}^{x_2} TC_{01,11,11}^{x_0,x_1,x_2}
\]

\[
(3)
\]

The \( 1 \rightarrow 0 \) transition can be studied in similar way. Eventually, the switching activity of an 3-input
AND gate can be calculated by:

\[ E(y) = p(y_{0\rightarrow1}) + p(y_{1\rightarrow0}) \]  

### 2.1.2. Computation of the Switching Activity of an n-input AND Gate

Based on the aforementioned analysis, the transition activity of an n-input AND gate, \( y=f(x_0, x_1, \ldots, x_{n-1}) = x_0x_1 \ldots x_{n-1} \), can be expressed by:

\[ E(y) = p(y_{0\rightarrow1}) + p(y_{1\rightarrow0}) \]

\[ = \frac{2^{n-2}}{n} \left( \prod_{i=0}^{n-2} \prod_{k=0}^{n-1} \prod_{l=k+1}^{n-1} B_{0\rightarrow1} \right) \]

\[ + \frac{2^{n-2}}{n} \left( \prod_{i=0}^{n-2} \prod_{k=0}^{n-1} \prod_{l=k+1}^{n-1} B_{1\rightarrow0} \right) \]  

where:

\[ A_{0\rightarrow1} = b_i(j)p_{1\rightarrow0}^{y} + (1 - b_i(j))p_{0\rightarrow1}^{y} \]

\[ B_{0\rightarrow1} = \{b_k(j)p_{1\rightarrow1}^{y} + (1 - b_k(j))b_l(j)\} \]

\[ = \sum_{n-2}^{2} \left( \prod_{i=0}^{n-2} \prod_{k=0}^{n-1} \prod_{l=k+1}^{n-1} A_{0\rightarrow1} \right) \]

\[ + \sum_{n-2}^{2} \left( \prod_{i=0}^{n-2} \prod_{k=0}^{n-1} \prod_{l=k+1}^{n-1} B_{1\rightarrow0} \right) \]  

\[ A_{1\rightarrow0} = b_i(j)p_{0\rightarrow1}^{y} + (1 - b_i(j))p_{1\rightarrow0}^{y} \]

\[ B_{1\rightarrow0} = \{b_k(j)p_{0\rightarrow1}^{y} + (1 - b_k(j))b_l(j)\} \]

\[ = \sum_{n-2}^{2} \left( \prod_{i=0}^{n-2} \prod_{k=0}^{n-1} \prod_{l=k+1}^{n-1} A_{1\rightarrow0} \right) \]

\[ + \sum_{n-2}^{2} \left( \prod_{i=0}^{n-2} \prod_{k=0}^{n-1} \prod_{l=k+1}^{n-1} B_{1\rightarrow0} \right) \]  

The term, \( b_i(j) \) corresponds to the \( i \)-th input of the \( j \)-th pair and equals to 0 when the \( i \)-th input changes and otherwise. The structure of the first part of Eq. (5) is depicted in Figure 1, where the segment \( A_{0\rightarrow1} \) expresses the contribution of the transition probabilities to the transition \( 0 \rightarrow 1 \), while the segment \( B_{0\rightarrow1} \) expresses the contribution of the TCCs. The new formulas, which calculates the switching activity of the n-input NAND, NOR, and OR gates are given in Appendix I.

### 2.1.3. Complexity

Equation (5) consists of two parts, the first of which corresponds to the transition \( 0 \rightarrow 1 \), while the second one to \( 1 \rightarrow 0 \). Since the two parts are identical, it is sufficient to study the transition \( 0 \rightarrow 1 \), only. The structure of the first part of Eq. (5) is depicted in Figure 1, where the segment \( A_{0\rightarrow1} \) expresses the contribution of the transition probabilities to the transition \( 0 \rightarrow 1 \), while the segment \( B_{0\rightarrow1} \) expresses the contribution of the TCCs.

\[ A_{0\rightarrow1} \] consists of \( n \) terms, while \( B_{0\rightarrow1} \) of \( \binom{n}{2} \) terms. For computation reasons, the segment \( B_{0\rightarrow1} \) is split into two sub-parts, \( i.e., B_{0\rightarrow1} \) and \( B_{0\rightarrow1} \), as it is shown in Figure 1. Specifically, the sub-part \( B_{0\rightarrow1} \) corresponds to TCCs, \( TC_{x_0,x_k}^{x_i,x_l} \), \( 0 \leq k < l \leq n-1 \), between the signal \( x_0 \) and the remaining
signals, while $B_{0\rightarrow l_b}$ includes the remaining TCCs, $TC_{k_i,b_i}$, $1 \leq k \leq l \leq n-1$. The computational complexity of each part of Eq. (5) in terms of multiplications, can be calculated by the following lemmas.

**Lemma 1** The computational complexity, $C_n(A_{0\rightarrow 1})$, of $A_{0\rightarrow 1}$ of an $n$-input AND gate in terms of multiplications equals:

$$C_n(A_{0\rightarrow 1}) = 2^{n+1} - 5 \quad (11)$$

**Proof** The part $A_{0\rightarrow 1}$ can be calculated recursively as explained in the following. In the beginning the sub-part of $A_{0\rightarrow 1}$ which corresponds to $(n-2)$ and $(n-1)$ bits is calculated performing $2^2$ multiplications and the result is stored $2^{n-2}$ times. Next, the sub-part that corresponds to $(n-3)$ to $(n-1)$ bits is calculated performing $2^3$ multiplications and the result is stored $2^{n-3}$ times and so on until the calculation the whole part $A_{0\rightarrow 1}$. Consequently, the required number of multiplications for the calculation of $A_{0\rightarrow 1}$ equals:

$$2^2 + 2^3 + \cdots + 2^n - 1 = 2^{n+1} - 5$$

**Lemma 2** The computational complexity, $C_n(B_{0\rightarrow 1})$, of $B_{0\rightarrow 1}$ of an $n$-input AND gate in terms of multiplications is:

$$C_n(B_{0\rightarrow 1}) = 2^{n+2} + 2^{n+1} - 8(n + 1) \quad \forall n \geq 3 \quad (12)$$

**Proof** The computational complexity of the segment $B_{0\rightarrow 1}$ can be computed as:

$$C_n(B_{0\rightarrow 1}) = C_n(B_{0\rightarrow l_a}) + C_n(B_{0\rightarrow l_b}) + C_n(Connect (B_{0\rightarrow l_a}, B_{0\rightarrow l_b})) \quad (13)$$

where $C_n(B_{0\rightarrow l_a})$ and $C_n(B_{0\rightarrow l_b})$ denote the complexities of the sub-parts $B_{0\rightarrow l_a}$ and $B_{0\rightarrow l_b}$, respectively. The last term corresponds to the required number of multiplications between $B_{0\rightarrow l_a}$ and $B_{0\rightarrow l_b}$. Then, the complexity of each term will be specified below:

(i) Using the approach of Lemma 2, the complexity $C_n(B_{0\rightarrow l_a})$ is calculated. That is:

$$C_n(B_{0\rightarrow l_a}) = 2^3 + 2^4 + \cdots + 2^n - 1 = 2^{n+1} - 9 \quad (14)$$

(ii) The complexity $C_n(B_{0\rightarrow l_b})$ is the complexity of an $(n-1)$-input gate. That is:

$$C_n(B_{0\rightarrow l_b}) = C_{n-1}(B_{0\rightarrow 1}) + 2 \quad (15)$$

(iii) To compute the whole segment $B_{0\rightarrow 1}$, the required multiplications are:

$$C_n(Connect (B_{0\rightarrow l_a}, B_{0\rightarrow l_b})) = 2^n - 1 \quad (16)$$

Substituting Eqs. (14), (15), and (16) into Eq. (13), we obtain
The complexity of Eq. (17) is computed by induction. We will prove that:

\[ C_n(B_{0-\rightarrow 1}) = 2^{n+2} + 2^n - 8(n+1) \quad \forall n \geq 3 \quad (18) \]

(i) For \( n = 3 \), Eq. (17) results into:

\[ C_3(B_{0-\rightarrow 1}) = \]

\[ 2^4 - 9 + [C_2(B_{0-\rightarrow 1}) + 2] \]

\[ + [2^3 - 1] = 16 \quad (19) \]

where \( C_{n-1}(B_{0-\rightarrow 1}) = 0 \), since a two-input gate includes one coefficient. It can be easily seen that Eq. (19) equals (18) for \( n = 3 \).

(ii) We assume that Eq. (18) holds for \( n = k \). That is:

\[ C_k(B_{0-\rightarrow 1}) = 2^{k+2} + 2^{k+1} - 8(k+1) \quad (20) \]

We will prove that Eq. (18) holds for \( n = k + 1 \).

Thus,

\[ C_{k+1}(B_{0-\rightarrow 1}) = C_{k+1}(B_{0-\rightarrow 1a}) + C_{k+1}(B_{0-\rightarrow 1b}) \]

\[ + C_{k+1}(\text{Connect}(B_{0-\rightarrow 1a}, B_{0-\rightarrow 1b})) = \]

\[ = [2^{k+2} - 9] + [C_k(B_{0-\rightarrow 1}) + 2] \]

\[ + [2^{k+1} - 1] = [2^{k+3} + 2^{k+2} - 8(k+2)] \]

Eventually,

\[ C_{k+1}(B_{0-\rightarrow 1}) = 2^{k+3} + 2^{k+2} - 8(k+2) \]

To make clear the previous description of the estimation method regarding to the number of multiplications, the switching estimation of a 4-input AND gate is examined below.

The calculation of the part \( A_{0-\rightarrow 1} \) is shown in Figure 2. In the beginning the sub-part that corresponds to \( x_2, x_3 \) inputs (bits \( n-2 \) to \( n-1 \)) is calculated and the partial product is stored four times (generally \( 2^{n-2} \) times). Afterwards, the sub-part that corresponds to \( x_1, x_2, x_3 \) inputs is calculated using the previous calculated partial

<table>
<thead>
<tr>
<th>Part ( A_{0-\rightarrow 1} )</th>
<th>Part ( B_{0-\rightarrow 1a} )</th>
<th>Part ( B_{0-\rightarrow 1b} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0</td>
<td>00 00 00 00</td>
</tr>
<tr>
<td>1</td>
<td>0 0 0 1</td>
<td>00 00 01</td>
</tr>
<tr>
<td>2</td>
<td>0 0 1 0</td>
<td>00 01 10</td>
</tr>
<tr>
<td>3</td>
<td>0 0 1 1</td>
<td>00 01 11</td>
</tr>
<tr>
<td>4</td>
<td>0 1 0 0</td>
<td>01 00 00</td>
</tr>
<tr>
<td>5</td>
<td>0 1 0 1</td>
<td>01 00 01</td>
</tr>
<tr>
<td>6</td>
<td>0 1 1 0</td>
<td>01 01 10</td>
</tr>
<tr>
<td>7</td>
<td>0 1 1 1</td>
<td>01 01 11</td>
</tr>
<tr>
<td>8</td>
<td>1 0 0 0</td>
<td>10 10 10</td>
</tr>
<tr>
<td>9</td>
<td>1 0 0 1</td>
<td>10 10 11</td>
</tr>
<tr>
<td>10</td>
<td>1 0 1 0</td>
<td>10 11 10</td>
</tr>
<tr>
<td>11</td>
<td>1 0 1 1</td>
<td>10 11 11</td>
</tr>
<tr>
<td>12</td>
<td>1 1 0 0</td>
<td>11 10 10</td>
</tr>
<tr>
<td>13</td>
<td>1 1 0 1</td>
<td>11 10 11</td>
</tr>
<tr>
<td>14</td>
<td>1 1 1 0</td>
<td>11 11 10</td>
</tr>
<tr>
<td>15</td>
<td>1 1 1 1</td>
<td>11 11 11</td>
</tr>
<tr>
<td>Number of multip.</td>
<td>27</td>
<td>23</td>
</tr>
</tbody>
</table>

FIGURE 2 Encoding of a 4 inputs AND gate for the switching activity computation.
product and the product is stored two times. At the end the whole \( A_{0-1} \) is calculated performing \( 2^{n-1} \) multiplications. The total required number of multiplications of \( A_{0-1} \) is 27 (i.e., \( C_{d}(A_{0-1}) = 2^{n+1} - 5 \)).

The part \( B_{0-1} \) is split into two sub-parts \( B_{0-1a} \) and \( B_{0-1b} \). As it shown in Figure 1, 23 multiplications are required for the calculation of \( B_{0-1a} \), while the part \( B_{0-1b} \) is the same with the \( B_{0-1} \) part of a 3-input AND gate shown in Figure 3.

Lemma 3 The computational complexity of an \( n \)-input AND gate, using the method [6] is:

\[
C_{n}(E(y)) = (2^{n} - 1)(n^{2} + n - 2)
\]

Proof It has been proved in [6] that the switching probability can be expressed as:

\[
p(y_{i-j}) = \sum_{\pi \in P_{i}} \sum_{\pi' \in P_{j}} \prod_{k=1}^{n} p(x_{k}) \sum_{1 \leq k \leq l \leq n} T_{C_{i-k-l}}^{x_{x_{i}} x_{i}}
\]

where \( P_{i} \) is the set of all paths of the corresponding OBDD in the ON set of \( f \), \( P_{j} \) is the set of all paths of the corresponding OBDD in the OFF set of \( f \), and \( x_{k}, y_{k} = 0, 1, 2 \) are the values of the variable \( x_{k} \) on the paths \( \pi \) and \( \pi' \), respectively. The number 2 denotes the don’t care state.

By traversing the paths of the OBDD of an \( n \)-input AND gate to calculate the \( p(y_{0-1}) \), the total number of products is equal to \( N = 2^{n} - 1 \). Also, the required multiplications of each product are:

\[
L = n + \binom{n}{2} - 1 = ((n^{2} + n - 2)/2).
\]

Eventually, the total number of the required multiplications of \( p(y_{0-1}) \) is \( N \times L \) and thus, \( C_{n}(p(y_{1-0})) = (2^{n} - 1)((n^{2} + n - 2)/2) \). Since \( E(y) = 2E(y_{1-0}) = 2E(y_{0-1}) \), it is obtained:

\[
C_{n}(E(y)) = (2^{n} - 1)(n^{2} + n - 2).
\]

Lemma 4 The complexity of the switching activity, \( E(sw) \), of the proposed method is always smaller than the complexity of the method [6].

Proof Since it holds:

\[
2^{n} = n^{2} + n > n^{2} - 15n
\]

\[
\Rightarrow n^{2} + n - 20 > n^{2} + n - 30 \Rightarrow 2^{n} > n^{2} + n - 30.
\]

From Eq. (21) and, (11) and (12), it is concluded that:

\[
(2^{n} - 1)(n^{2} + n - 2) > 2^{n+1} + 2^{n} - [8(n + 1) - 6]
\]

\[
\Rightarrow 2^{n}(n^{2} + n - 20) > n^{2} - 15n - 30.
\]

Table I shows the computational complexity of the proposed method and [6] in terms of multiplications for \( n=2,3,\ldots,10 \) as well as the associated reductions. It can be noticed that the larger \( n \), the larger the reduction.

<table>
<thead>
<tr>
<th>Word number</th>
<th>( x_{0}, x_{1}, x_{2} )</th>
<th>( T_{C_{h_{1}:h_{4}}}^{x_{h_{1}} x_{h_{4}}} )</th>
<th>( T_{C_{h_{2}:h_{3}}}^{x_{h_{2}} x_{h_{3}}} )</th>
<th>( T_{C_{h_{3}:h_{2}}}^{x_{h_{3}} x_{h_{2}}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>1</td>
<td>0 0 1</td>
<td>00</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0</td>
<td>01</td>
<td>00</td>
<td>01</td>
</tr>
<tr>
<td>3</td>
<td>1 0 0</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>4</td>
<td>1 0 1</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>1 1 0</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>1 1 1</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>1 1 1</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>Number of multiplications</td>
<td>11</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure 3 Encoding of a 3-inputs AND gate for the switching activity computation.
TABLE I Complexity of the switching activity of an n-input basic gate

<table>
<thead>
<tr>
<th>No. of multiplications for ( E(y) ), ( n \geq 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L = \text{p}(x_{k_{i,j} = 0}) )</td>
</tr>
<tr>
<td>( M = \text{TC}<em>{k</em>{i,j} = 0} )</td>
</tr>
<tr>
<td>( \text{L times M} )</td>
</tr>
<tr>
<td>( n \geq 3 )</td>
</tr>
<tr>
<td>( 12, \ n = 2 )</td>
</tr>
<tr>
<td>( )</td>
</tr>
<tr>
<td>( \text{Proposed approach} )</td>
</tr>
<tr>
<td>( \text{Approach [6]} )</td>
</tr>
</tbody>
</table>

2.1.4. Accuracy of the Proposed Method

**Lemma 5** The proposed method exhibits the same accuracy with method [6].

**Proof** It is sufficient to prove that the proposed method and [6] result into identical transition probabilities. For that purpose a 3-input AND gate is considered. Employing Eq. (22), the switching probability \( p(y_{0-1}) \) can be computed as follows:

\[
p(y_{0-1}) = p_{0,0} p_{1,1} p_{1,1} + p_{1,0} p_{0,1} p_{1,0} + p_{0,1} p_{1,0} p_{0,1} + p_{1,0} p_{0,1} p_{0,1} + p_{0,1} p_{1,0} p_{0,1} + p_{1,0} p_{0,1} p_{0,1} + p_{0,1} p_{1,0} p_{0,1} + p_{1,0} p_{0,1} p_{0,1} + p_{0,1} p_{1,0} p_{0,1} + p_{1,0} p_{0,1} p_{0,1} + p_{0,1} p_{1,0} p_{0,1} + p_{1,0} p_{0,1} p_{0,1}
\]

Substituting the don't care state (i.e., subscript 2) with the values \( \{0, 1\} \), Eq. (23) can be re-written as:

\[
p(y_{0-1}) = \frac{p_{0,0} p_{1,1} p_{1,1}}{p_{0,1} p_{0,1} p_{1,0} + p_{0,1} p_{0,1} p_{1,0} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1} + p_{0,1} p_{0,1} p_{0,1}}
\]

It is concluded that Eqs. (5) and (24) are identical.

2.2. Switching Activity Computation of a Logic Network

Taking into account all types of correlation, the switching activity calculation of any gate output, needs the TCCs of its inputs. However, only the TCCs of the primary inputs of the logic network are provided. Consequently, a mechanism for the calculation and propagation of the TCCs through the wires of all circuit levels must be developed. First, the theoretical framework for the calculation and propagation of the TCCs is given. Then, the implemented method for the propagation of these coefficient is also presented. It should be stressed that the proposed method based on a prepossessing of the logic network, achieves to propagate only the required TCCs and therefore, significant savings can be achieved.

2.2.1. Propagation of the Transition Correlation Coefficients

It has been proved [6] that the correlation coefficient between a signal \( x \) and node \( f \) with immediate inputs \( x_0, x_1, \ldots, x_{n-1} \), from which at least one depends on the signal \( x \), is:

\[
TC_{x,f} = \frac{p(f_{i,j} \wedge x_{p-q})}{p(f_{i,j}) p(x_{p-q})}
\]

where \( i, j, p, q \in \{0, 1\} \).

Since the transition probabilities of the node \( f \) and signal \( x \) have been already computed, the problem is reduced to the computation of \( p(f_{i,j} \wedge x_{p-q}) \). Also, \( TC_{x,f} \) is computed by [6]:

\[
TC_{x,f} = \left( \sum_{p=0}^{2^n} \sum_{p=0}^{2^n} \prod_{k=1}^{n} TC_{x,f_{k_{p,k} = 0}} \right) / p(f_{i,j})
\]

It can be seen that the numerator of Eq. (26) consists of: (i) the product of \( TC_{x,f_{p,k,j} = 0} \)'s, (ii) the product of \( p(x_{k_{p,k} = 0}) \)'s, and (iii) the product of the \( TC_{x,f_{p,k,j} = 0} \)'s. Comparing Eq. (26) with (5), we infer that both equations similar structure except the first part. Consequently, the computational complexity in terms of multiplications is the complexity of...
Eq. (5) plus \((2^n - 1)(n - 1)\) multiplications due to the first part.

**Lemma 6** Let \(f\) be the output of an \(n\)-input AND gate and \(x\) be an arbitrary signal of a logic network. The number of the required multiplications for calculating of one TCC between two signals \(f\) and \(x\) is computed by:

\[
C_{\text{TCC}}(n) = (2^n - 1)(n^2 + 3n - 2)
\]

**Proof** Employing the approach of Lemma 3, we infer that \(n\) additional multiplications are required, since there exist \(n\) additional TCCs, \(TC_{T_{\mu},T_{\mu}}^{x,y}\), that is \(R = n + n + \left(\frac{\sqrt{3}}{2}\right) - 1 = (n^2 + 3n - 2)/2\). Therefore, the total number of multiplications is \(N \times R\). Eventually, \(C_{\text{TCC}}(n) = p(y_{0-1}) + p(y_{1-0}) = (2^n - 1)(n^2 + 3n - 2)\). Identical results can be derived if \(f\) is the output of one of the remaining basic gates.

Table II summarises the computational complexity of the proposed method and Marculescu's one in terms of the multiplications. It must be noticed that the Table II expresses the computational complexity of the TCs when the node \(f\) performs a \(1 \rightarrow 0\) or a \(0 \rightarrow 1\) transition (i.e., \(i, j=0,1\) or \(i, j=1,0\)). Since \(TC_{T_{\mu},T_{\mu}}^{x,y}\) with \(i, j=0,1\) or \(i, j=1,0\) and \(p, q \in \{0,1\}\), this case covers 8 of the 16 coefficients.

In case of a transition between the non-controlling values of a gate with output \(f\), for instance an AND gate, it means that \(i, j=1,1\) and the required number of multiplications is equal to \((n-1)+(n-1)+\left(\frac{(n-1)}{2}\right) = (n^2 + 3n)/2\). The paths \(\pi\) and \(\pi'\) are identical and due to the fact that are reached at the non-controlling value of the gate, only one path exists. Therefore, the part \(C\) consists of \(n\) terms and requires \(n-1\) multiplications, and the part \(A\) consists of \(n\) terms and requires \(n-1\) multiplications, and the part \(B\) consists of \(\left(\frac{\sqrt{3}}{2}\right)\) terms and requires \((\left(\frac{\sqrt{3}}{2}\right)-1)\) multiplications. Eventually, two additional multiplications are required for the connection of the three parts. This case covers 4 of the 16 coefficients since \(TC_{T_{\mu},T_{\mu}}^{x,y}\), \(i, j=\text{"non controlling value"},\) and \(p, q \in \{0,1\}\).

The four remaining TCCs stem from the controlling values of a gate. Up to now the first 12 TCCs have been already calculated. However, it has been proved [6] that the system of the sixteen TCCs can be solved when at least 9 of the 16 coefficients are known.

**Lemma 7** The complexity of the propagation of TCCs of the proposed method is always less than the corresponding complexity of [6] when \(f_{0-1}\) or \(f_{1-0}\).

**Proof** Due the fact that the additional multiplications, \((2^n-1)(n-1)\), of Eq. (29) is added to the complexity of both methods and using Lemma 6, we conclude that the proposed method has less complexity.

| Table II Complexity for the propagation of one TC, in terms of multiplications |
|---------------------------------|---------------------------------|
| Number of multiplications for \(TC_{T_{\mu},T_{\mu}}^{x,y}\), \(n \geq 2\) |                                    |
| \(L = p(x_{k,a\rightarrow})\) | \(2^n + 1 - 5\) | \(n - 1\) |
| \(M = TC_{T_{\mu},T_{\mu}}^{x,y}\) | \(0, n = 2\) | 
| \(K = TC_{T_{\mu},T_{\mu}}^{x,y}\) | \((2^n - 1)(n - 1)\) | \(n - 1\) |
| \(L \times M\) | \(2^n - 1\) | 1 |
| \(K \times L\) | \(2^n - 1\) | 1 |
| Proposed approach | \(24, n = 2\) |                                    |
| Approach [6] | \((2^n - 1)(2n + \left(\frac{\sqrt{3}}{2}\right) - 1)\) | \((2n + \left(\frac{\sqrt{3}}{2}\right) - 1)\) |
2.2.2. The Proposed Algorithm

The proposed algorithm performs: (i) the circuit pre-processing, (ii) the calculation and propagation of TCCs, and (iii) the calculation of the switching activity of any node. In particular, the circuit pre-processing is a forward traversal from the primary inputs to the primary outputs and consists of the steps:

(i) Identification of the gate type: We specify the type of each basic gate and thus, the proper formula for the switching activity calculation.

(ii) Levelization of the circuit: We specify the logic level of the logic network in which a gate belongs to.

(iii) Correlation Level Length: The Correlation Level Length of a node $g$, $CLL_g$, specifies the number of the logic levels, whose signal correlations are considered for calculating node $g$ switching activity. Apparently, if two signals $x_0$ and $x_1$ have $CLL_g$ greater than a chosen $CLL$, the signals are uncorrelated, that is $TC_{x_0,x_1}^{i,j,p,q} = 1$.

(iv) Correlation List: Choosing a certain $CLL_g$, the Correlation List of a node $g$, $CL_g$, specifies the correlated signal pairs.

The structure of the developed algorithm is:

\begin{verbatim}
function power_estimation (F, X)
  begin
    for each gate $g \in F$:
      1. find the type of gate $g$
      2. find the level of gate $g$
      3. construct the dependence list $L_g$
      4. mark the proper TCCs of the pairs in $L_g$
    for each level $i$
      for each gate $g \in F$: calculate the transition probability $E(g)$
      for each pair of signals of level $i$: pre-compute the marked TCCs
    return switching activities $E(g)$ for each node $g$
  end function power_estimation;
\end{verbatim}

Example The application of the previous algorithm is illustrated by a certain logic network shown in Figure 4. Our main purpose is to show the reduction of the computational complexity using the Correlation List.

The logic network shown in Figure 4 consists of: 6 primary inputs, 2 primary outputs, and 6 levels. The level and correlation list if $CLL=2$, of the node $x_{11}$ are: $level(x_{11}) = \max\{Level(x_3), Level(x_{10})\} + 1 = \max\{3, 2\} + 1 = 4$ and $CL_{x_{11}} = \{x_1, x_{10}, x_{7}, x_4, x_5\}$, respectively. Furthermore, the contents of the Correlation List, $CL_{x_{11}}$, for a full transition of the node $x_{11}$ will be found. The associated probability in the case node $x_{11}$ performs a $1 \rightarrow 0$ transition is
given by:
\[ E_{1\rightarrow 0}(x_{11}) = P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,01}}^{x_{10},x_{10}} + \\
+ P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,10}}^{x_{10},x_{10}} + \\
+ P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,10}}^{x_{10},x_{10}} \]
(28)

Since \( CLL = 2 \) the appropriate pairwise correlation coefficients between the signals \((x_7, x_{10})\) and signals \((x_8, x_{10})\) must be computed. Taking into account the Eq. (26) the numerator which corresponds to the \( T_{C_{11,01}}^{x_{9},x_{10}} \) is equal to:
\[
\left\{ \begin{array}{l}
T_{C_{11,01}}^{x_7,x_{10}} T_{C_{11,01}}^{x_8,x_{10}} P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,00}}^{x_{7},x_{10}} + \\
T_{C_{11,10}}^{x_7,x_{10}} T_{C_{11,10}}^{x_8,x_{10}} P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,00}}^{x_{7},x_{10}} + \\
T_{C_{11,10}}^{x_7,x_{10}} T_{C_{11,10}}^{x_8,x_{10}} P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,10}}^{x_{7},x_{10}} 
\end{array} \right. 
(29)
\]
while the corresponding numerator of \( T_{C_{11,10}}^{x_{9},x_{10}} \) is equals to:
\[
T_{C_{11,10}}^{x_7,x_{10}} T_{C_{11,10}}^{x_8,x_{10}} P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,11}}^{x_{7},x_{8}} \]
(30)
and eventually, the corresponding numerator of \( T_{C_{11,00}}^{x_{9},x_{10}} \) is equals to:
\[
\left\{ \begin{array}{l}
T_{C_{11,00}}^{x_7,x_{10}} T_{C_{11,00}}^{x_8,x_{10}} P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,00}}^{x_{7},x_{10}} + \\
T_{C_{11,10}}^{x_7,x_{10}} T_{C_{11,10}}^{x_8,x_{10}} P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,00}}^{x_{7},x_{10}} + \\
T_{C_{11,10}}^{x_7,x_{10}} T_{C_{11,10}}^{x_8,x_{10}} P_{I_{10}}^x P_{I_{10}}^x T_{C_{11,10}}^{x_{7},x_{10}} 
\end{array} \right. 
(31)
\]
Finally, the necessary correlation coefficients between the signals \((x_7, x_{10})\) and \((x_8, x_{10})\) can be summarized as follows:
\[
CL_{1_{x_{10}}} = \{ T_{C_{11,01}}^{x_7,x_{10}}, T_{C_{11,11}}^{x_7,x_{10}}, T_{C_{11,10}}^{x_7,x_{10}}, T_{C_{11,00}}^{x_7,x_{10}} \} 
\]
(32)
\[
CL_{2_{x_{10}}} = \{ T_{C_{11,01}}^{x_8,x_{10}}, T_{C_{11,11}}^{x_8,x_{10}}, T_{C_{11,10}}^{x_8,x_{10}}, T_{C_{11,00}}^{x_8,x_{10}} \} 
\]
(33)
Following a similar analysis for the \( 0 \rightarrow 1 \) transitions it can be found that:
\[
CL_{3_{x_{11}}} = \{ T_{C_{11,01}}^{x_7,x_{10}}, T_{C_{11,11}}^{x_7,x_{10}}, T_{C_{11,10}}^{x_7,x_{10}} \} 
\]
(34)
\[
CL_{4_{x_{11}}} = \{ T_{C_{11,01}}^{x_8,x_{10}}, T_{C_{11,11}}^{x_8,x_{10}}, T_{C_{11,10}}^{x_8,x_{10}} \} 
\]
(35)
Eventually,
\[
CL_{x_{11}} = \bigcup_{i=1}^{4} CL_{x_{ii}} 
\]
(36)
Therefore, the required number of correlation coefficients is a subset of the sixteen coefficients of the each pair \((x_7, x_{10})\) and \((x_8, x_{10})\). Hence the above example shows the usefulness of the Correlation List in the reduction of the computational complexity.

3. RESULTS

The proposed method was implemented by C language and the experiments are performed on HP 735 workstation with 64 MB of memory.

Table III describes the characteristics of the benchmark circuits in terms of the number of the primary inputs, \( n \), the primary outputs, \( m \), the logic levels, \( L \), and the number of gates.

<table>
<thead>
<tr>
<th>Circuit # inputs</th>
<th># outputs</th>
<th># gates</th>
<th># levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>5</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>C432</td>
<td>32</td>
<td>7</td>
<td>184</td>
</tr>
<tr>
<td>C499</td>
<td>41</td>
<td>32</td>
<td>176</td>
</tr>
<tr>
<td>C3880</td>
<td>60</td>
<td>26</td>
<td>221</td>
</tr>
<tr>
<td>C1355</td>
<td>41</td>
<td>32</td>
<td>180</td>
</tr>
<tr>
<td>C1908</td>
<td>33</td>
<td>25</td>
<td>205</td>
</tr>
<tr>
<td>C3540</td>
<td>50</td>
<td>22</td>
<td>1007</td>
</tr>
<tr>
<td>C6288</td>
<td>32</td>
<td>32</td>
<td>1491</td>
</tr>
<tr>
<td>alu4</td>
<td>14</td>
<td>8</td>
<td>547</td>
</tr>
<tr>
<td>z4ml</td>
<td>7</td>
<td>4</td>
<td>98</td>
</tr>
<tr>
<td>duke2</td>
<td>22</td>
<td>29</td>
<td>306</td>
</tr>
</tbody>
</table>

Table IV and V show the computational complexity of the proposed method and the approach of [6] in terms of the additions and multiplications for certain ISCAS circuits, considering \( CLL = 3 \) and \( CLL = 6 \), respectively. Column 2 gives the number, \( S \), of the pairs of the correlated signals. Column 3 gives the required number of additions, which is the same for the two methods. Column 6 gives the percentage of the multiplication reduction. It must be noticed that the larger the Correlation Level Length, the larger the number of the correlated signals, since more pairs of signals are correlated.

In Table VI, power estimation results for seven large ISCAS benchmark circuits for a permitted
level of correlation $CLL = 3$ are presented. For these results a pseudo-random input vector set is used and the absolute error in a node-by-node comparison is reported. As it is shown, the MAX, MEAN, RMS and STD errors are small enough even for this small level of $CLL$.

Finally in Table VII, power estimation results with $CLL = 3$ are reported, using as input vector set a highly correlated input vector from a binary counter sequence. The reported errors are higher than the errors that are reported in previous Table VI, but they are still acceptable. All the values of power consumption are in $\mu W$ at 20 MHz and 5 Volts.

The power estimation tool, which has been used to obtain the results of Tables VI and VII, is an

<table>
<thead>
<tr>
<th>Circuit</th>
<th>$S$</th>
<th>1, 2 (+)</th>
<th>1 (+)</th>
<th>2 (+)</th>
<th>Gain %</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>19</td>
<td>524</td>
<td>464</td>
<td>512</td>
<td>+9.38</td>
<td>0.02</td>
</tr>
<tr>
<td>C432</td>
<td>2152</td>
<td>1070790</td>
<td>3555780</td>
<td>12041100</td>
<td>+70.47</td>
<td>38.08</td>
</tr>
<tr>
<td>C499</td>
<td>2472</td>
<td>197150</td>
<td>3252930</td>
<td>5837080</td>
<td>+44.27</td>
<td>36.31</td>
</tr>
<tr>
<td>C880</td>
<td>2052</td>
<td>174166</td>
<td>4577050</td>
<td>+47.61</td>
<td>30.24</td>
<td></td>
</tr>
<tr>
<td>C432</td>
<td>6909</td>
<td>2807330</td>
<td>11093600</td>
<td>31525500</td>
<td>+64.81</td>
<td>118.44</td>
</tr>
<tr>
<td>C499</td>
<td>6913</td>
<td>536762</td>
<td>699713</td>
<td>+9.69</td>
<td>9.71</td>
<td></td>
</tr>
<tr>
<td>C880</td>
<td>6234</td>
<td>396978</td>
<td>824143</td>
<td>+21.69</td>
<td>9.7</td>
<td></td>
</tr>
<tr>
<td>C1355</td>
<td>5667</td>
<td>472046</td>
<td>600483</td>
<td>+12.99</td>
<td>8.83</td>
<td></td>
</tr>
<tr>
<td>C1908</td>
<td>5036</td>
<td>831122</td>
<td>2822290</td>
<td>+38.68</td>
<td>43.1</td>
<td></td>
</tr>
<tr>
<td>C3540</td>
<td>29932</td>
<td>5343370</td>
<td>38926700</td>
<td>+59.46</td>
<td>176.61</td>
<td></td>
</tr>
<tr>
<td>C6288</td>
<td>5524</td>
<td>5280000</td>
<td>28222900</td>
<td>+59.46</td>
<td>176.61</td>
<td></td>
</tr>
<tr>
<td>Alu4</td>
<td>2947</td>
<td>299674</td>
<td>527321</td>
<td>+46.72</td>
<td>7.09</td>
<td></td>
</tr>
<tr>
<td>Z4ml</td>
<td>199</td>
<td>10126</td>
<td>10482</td>
<td>+8.3</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>Duke2</td>
<td>5524</td>
<td>5280000</td>
<td>28222900</td>
<td>+59.46</td>
<td>176.61</td>
<td></td>
</tr>
</tbody>
</table>

(1) Proposed method, (2) [6]; (+) number of additions; (,) number of multiplications.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>$S$</th>
<th>1, 2 (+)</th>
<th>1 (+)</th>
<th>2 (+)</th>
<th>Gain %</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>19</td>
<td>524</td>
<td>464</td>
<td>512</td>
<td>+9.38</td>
<td>0.02</td>
</tr>
<tr>
<td>C432</td>
<td>2152</td>
<td>1070790</td>
<td>3555780</td>
<td>12041100</td>
<td>+70.47</td>
<td>38.08</td>
</tr>
<tr>
<td>C499</td>
<td>2472</td>
<td>197150</td>
<td>3252930</td>
<td>5837080</td>
<td>+44.27</td>
<td>36.31</td>
</tr>
<tr>
<td>C880</td>
<td>2052</td>
<td>174166</td>
<td>4577050</td>
<td>+47.61</td>
<td>30.24</td>
<td></td>
</tr>
<tr>
<td>C432</td>
<td>6909</td>
<td>2807330</td>
<td>11093600</td>
<td>31525500</td>
<td>+64.81</td>
<td>118.44</td>
</tr>
<tr>
<td>C499</td>
<td>6913</td>
<td>536762</td>
<td>699713</td>
<td>+9.69</td>
<td>9.71</td>
<td></td>
</tr>
<tr>
<td>C880</td>
<td>6234</td>
<td>396978</td>
<td>824143</td>
<td>+21.69</td>
<td>9.7</td>
<td></td>
</tr>
<tr>
<td>C1355</td>
<td>5667</td>
<td>472046</td>
<td>600483</td>
<td>+12.99</td>
<td>8.83</td>
<td></td>
</tr>
<tr>
<td>C1908</td>
<td>5036</td>
<td>831122</td>
<td>2822290</td>
<td>+38.68</td>
<td>43.1</td>
<td></td>
</tr>
<tr>
<td>C3540</td>
<td>29932</td>
<td>5343370</td>
<td>38926700</td>
<td>+59.46</td>
<td>176.61</td>
<td></td>
</tr>
<tr>
<td>C6288</td>
<td>5524</td>
<td>5280000</td>
<td>28222900</td>
<td>+59.46</td>
<td>176.61</td>
<td></td>
</tr>
<tr>
<td>Alu4</td>
<td>2947</td>
<td>299674</td>
<td>527321</td>
<td>+46.72</td>
<td>7.09</td>
<td></td>
</tr>
<tr>
<td>Z4ml</td>
<td>199</td>
<td>10126</td>
<td>10482</td>
<td>+8.3</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>Duke2</td>
<td>5524</td>
<td>5280000</td>
<td>28222900</td>
<td>+59.46</td>
<td>176.61</td>
<td></td>
</tr>
</tbody>
</table>

(1) Proposed method, (2) [6]; (+) number of additions; (,) number of multiplications.
in-house developed tool. It consists of the following components:

(i) a library of primitive gates used to perform the mapping of the circuit,
(ii) a vector file (text file) corresponding to the applied input vector set,
(iii) a capacitance file, which provides the capacitance information of every circuit node. In case of the capacitance file is not available an estimation is used considering the fanout of each node,
(iv) the proposed switching activity estimator implemented in C.

A SLIF format or structural VHDL can be used to describe the circuit. More details on this tool can be found in: [14].

4. CONCLUSIONS

An efficient method for switching activity estimation of logic level networks is proposed. The method had as inputs the static and transition probabilities, the data correlations, and the exact structure of the logic network. New formulas for calculating the switching activity of a single gate, which are used by a novel algorithm for estimating the activity of any node, are proposed. The method provides accurate switching activity values, with small time cost. The experimental results proved that significant reduction of the required multiplications can be achieved.

References


APPENDIX I

(A) The transition activity of an $n$-input NAND gate is given by:

$$E(y) = \sum_{j=0}^{2^n-2} \left( \prod_{l=0}^{n-2} A_{1-l} \prod_{k=0}^{n-2} B_{1-k} \right)$$

where:

$$A_{1-l} = b_l(j) p_{X_l}^{1-l} + (1 - b_l(j)) p_{X_l}^{0-l}$$

$$B_{1-k} = \{ b_k(j) b_j(j) T_{C_{11,11},X_l} + (1 - b_k(j)) b_j(j) T_{C_{11,01},X_l} \}$$

$$(11)$$
The transition activity of an \( n \)-input OR gate is given by:

\[
E(y) = \sum_{j=0}^{2^n-2} \left( \prod_{i=0}^{n-2} A_{j-i} \prod_{k=0}^{n-1} B_{1-i-k} \right) + \sum_{j=0}^{2^n-2} \left( \prod_{i=0}^{n-2} A_{j-i} \prod_{k=0}^{n-1} B_{0-i-k} \right)
\]

where:

\[
A_{1-i} = b_i(j) p_{0-i-0} + (1 - b_i(j)) p_{0-i-1}
\]

\[
B_{1-i} = \{b_k(j) b_l(j) T_{0-i-i} + (1 - b_k(j)) b_l(j) T_{0-i-0} + b_k(j) (1 - b_l(j)) T_{0-i-1} + (1 - b_k(j) (1 - b_l(j)) T_{0-i-0} \}
\]

(C) The transition activity of an \( n \)-input NOR gate is given by:

\[
E(y) = \sum_{j=0}^{2^n-2} \left( \prod_{i=0}^{n-2} A_{j-i} \prod_{k=0}^{n-1} B_{1-i-k} \right) + \sum_{j=0}^{2^n-2} \left( \prod_{i=0}^{n-2} A_{j-i} \prod_{k=0}^{n-1} B_{0-i-k} \right)
\]

Authors' Biographies

George Theodoridis received his Diploma in Electrical Engineering from the University of Patras, Greece, in 1994. Since then, he is currently working towards to Ph.D. at Electrical Engineering, University of Patras. His research interests include low power design, logic synthesis, computer arithmetic, and power estimation.

Spyros Theoharis received his Diploma in Computer Engineering and Informatics from the University of Patras, Greece, in 1994. Since then, he is currently working towards to Ph.D. at Electrical Engineering, University of Patras. His research interests include low power design, multilevel logic synthesis, parallel architectures, and power estimation.

Dimitrios Soudris received his Diploma in Electrical Engineering from the University of Patras, Greece, in 1987. He received the Ph.D. Degree from in Electrical Engineering, from the University of Patras in 1992. He is currently working as Ass. Professor in Dept. of Electrical and Computer Engineering, Democritus University of Thrace, Greece. His research interests include parallel architectures, computer arithmetic, vlsi signal processing, and low power design. He has published more than 50 papers in international
Costas Goutis was a Lecturer at School of Physics and Mathematics at the University of Athens, Greece, from 1970 to 1972. In 1973, he was Technical Manager in the Greek P.P.T. He was Research Assistant and Research fellow in the Department of Electrical Engineering at the University of Strathclyde, U.K., from 1976 to 1979, and Lecturer in the Department of Electrical and Electronic Engineering at the University of Newcastle upon Tyne, U.K., from 1979 to 1985. Since he has been Associate Professor and Full Professor in the Department of Electrical and Computer Engineering, University of Patras, Greece. His recent interests focus on VLSI Circuit Design, Low Power VLSI Design, Systems Design, Analysis and Design of Systems for Signal Processing and Telecommunications. He has published more than 150 papers in international journals and conferences. He has been awarded a large number of Research Contracts from ESPRIT, RACE, and National Programs.
Submit your manuscripts at http://www.hindawi.com