*VLSI DESIGN*
1999, Vol. 9, No. 2, pp. 147–157
Reprints available directly from the publisher
Photocopying permitted by license only



# A New Method for Low Power Design of Two-Level Logic Circuits

G. THEODORIDIS[a,*], S. THEOHARIS[a], D. SOUDRIS[b] and C. GOUTIS[a]

[a] *VLSI Design Laboratory, Dept. of Electrical and Computer Engineering, University of Patras, Rio 26 110, Greece;*
[b] *VLSI Design and Testing Center, Laboratory of Electrical & Electronic Materials Technology, Dept. of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi 67 100, Greece*

*(Received 24 March 1997; In final form 18 January 1998)*

A new method for implementing two-level logic circuits, which exhibit minimal power dissipation, is introduced. Switching activity reduction of the logic network nodes is achieved by adding extra input signals to specific gates. Employing the statistic properties of the primary inputs, a new concept for grouping the input variables with similar features is introduced. Appropriate input variables are chosen for reducing the switching activity of a logic circuit. For that purpose, an efficient synthesis algorithm, which generates the set of all groups of the variables and solves the minimum covering problem for each group is developed. The comparison of the results, produced by the proposed method, and those from ESPRESSO shows that a substantial power reduction can be achieved.

*Keywords:* Low power design, logic circuit, power estimation, switching activity, blocking variable, temporal correlation

## 1. INTRODUCTION

The modern design techniques require to take into consideration in addition to the two traditional design parameters, area and speed, a third one, power dissipation [1, 2, 3]. The wide spread use of portable and wireless communication systems increases the demand for the extension of battery life. Because the current advances in battery technology are insufficient as the applications require, low-power design of integrated circuits becomes a challenging problem [1, 2, 3]. Also, reasons related with energy problems oblige the designers to tend at systems that dissipate less energy.

The power dissipation of a CMOS circuit consists of three factors, namely the static, the short-circuit, and the dynamic power dissipation. The prominent factor is the last one and is directly related with the switching activity of the circuit.

* Corresponding author. Tel.: + 30 61 997283, Fax: + 30 61 994798, e-mail: theodor@ee.upatras.gr

Different logic implementations of the same Boolean function result in circuits with different switching activities.

Applying the well-known CAD tools ESPRESSO [4, 5, 6], ESPRESSO Exact [5, 7], and ESPRESSO Signatures [5, 8] two-level logic synthesis can be performed. All the above techniques aim to the optimization of a cost function related with the area minimization. Recently new methods for minimizing the power dissipation of a logic function have been presented in [9, 10, 11, 12]. In particular, Shen *et al.* [9] presented a method where the power cost of a function of the primes was used to solve an ESPRESSO-like procedure. This procedure did not, however, consider including non-prime implicants in the cover although these implicants can lead to more power reduction. In [10], Vrudhula and Xie proposed a method for two-level implementations called modified-EXPAND, using the statistic characteristics of the signals. However, it has been assumed that all input signals: (i) have static probability equal to 0.5, (ii) are not spatiotemporally correlated, and (iii) their corresponding power dissipation is not taken into account. In contrast, Iman and Pedram in [11, 12] considered signals with random probabilities and took the transitions of the primary input signals into account. Also, they defined a new notation called Power Prime Implicants (PPI). These PPI's specify the set of all implicants that are sufficient and necessary for obtaining a minimum power solution. Based on the static probabilities of the function variables, they proposed an algorithm that generates all the PPIs, which are necessary to solve the minimum covering problem. Power savings up to 4% can be achieved. However, it is assumed that all signals are not spatiotemporally correlated and therefore, inaccurate power estimation takes place.

In this paper a new method for implementing low power two-level logic circuits using the static and transition probabilities of the inputs signals (*i.e.*, temporal correlation), is introduced. The temporal correlation has been chosen to be studied first in our research, among the remaining types of correlations, since this type of correlation occurs *always* in all input signals. Without loss of generality, if we have random signals, it can be considered that there is *no* spatial correlation and thus, the temporal correlation has the dominant role in power dissipation. In other words, the temporal correlation does not depend on the configuration of the input signals, while the spatial correlation does on.

The proposed work embodies fundamental principles from the ESPRESSO environment. Switching activity reduction of the logic network nodes is achieved by adding additional input signals to specific gates. These signals force the first level gate outputs in logic level zero for a number of combinations of the input signals. A formal method for the determination and classification of these signals is described. A synthesis tool for designing two-level logic circuits in terms of power, which selects the appropriate input variables and uses the main routines of ESPRESSO is implemented. Given the user-specified average power consumption per gate and the number of the partitions of the whole boolean space, a circuit with minimal power consumption can be achieved. The experimental results for different statistic features of the primary inputs and logic circuits indicate that proposed method achieves substantial power savings.

This paper is organized as follows: In Section 2 the power model is presented. Starting from an example, the proposed method is described in Section 3. Results and conclusions are presented in Sections 4 and 5, respectively.

## 2. POWER DISSIPATION MODEL

The dynamic power dissipation of a digital CMOS circuit can be expressed by the total number of transitions of its nodes. The dynamic power dissipation for an intermediate gate is given by:

$$P_{\text{gate}} = \frac{1}{2} \sum_{x_i \in \text{gate inputs}} C_{x_i} V^2 f E(x_i) \qquad (1)$$

where $C_{x_i}$ the capacitance of the $i$-th input, $V$ is the operation voltage, $f$ is the clock frequency, and $E(x_i)$ the transition probability of the $i$-th input. Hence, the total dynamic power dissipation of a complex logic circuit consisting of cascaded gates, is calculated by:

$$P_{\text{tot}} = P_{\text{out}} + \sum P_{\text{gate}} \tag{2}$$

where $P_{\text{out}}$ is the dynamic power dissipation consumed for charging and discharging the output load.

A signal $x_i$ of the logic function, $y = f(\underline{x}) = f(x_1, x_2, \ldots, x_n)$, can be described precisely with two attributes, namely the *static probability*, $p(x_i)$, and the *transition probability*, $E(x_i)$ [11, 12, 13]. More specifically, the static probability is defined as the percentage of the time in which the signal is high and, the transition probability is defined as the probability of the signal to make a transition during two successive clock cycles.

Adopting the power model proposed in [13], it is assumed that the primary inputs are mutually-independent and each primary input is first-order temporally-dependent. Given a logic function $y = f(\underline{x}) = f(x_1, x_2, \ldots, x_n)$, the associated Transition function ($Tf$) [13] is defined as:

$$Tf = f(\underline{x}^0) \oplus f(\underline{x}^T) \tag{3}$$

where $f(\underline{x}^0)$ and $f(\underline{x}^T)$ are the values of the function at the time instances $t = 0$ and $t = T$, respectively.

The Transition function depends on the primary inputs and takes the logic value one if and only if a transition occurs between two successive clock cycles. Therefore, the required transition probability, $E(y)$, of the gate output, $y = f(\underline{x})$, can be calculated by $E(y) = p(Tf = 1)$. Based on the Shannon expansion and Markov chain theory, the probability Transition function can be calculated by a recursive technique [13].

We deal with logic functions implemented by two-level circuits, where the first level consists of AND gates and the second one consists of an OR gate. Based on [13] the calculation of the transition probability of a $n$-input AND gate, $y = x_1 x_2 \ldots n_n$, is expressed as:

$$E(y) = \sum_{m=2}^{n} \left\{ \sum_{j_{m-1}=1} \sum_{j_m=j_{m-1}+1} \sum_{j_{m+1}=j_m+1} \cdots \sum_{j_{n-1}=j_{n-2}+1} \prod_{\substack{i=1 \\ i \neq j_z \forall z}}^{n} \cdot \prod_{z=m-1}^{n-1} 2p_{10}(x_i) p_{11}(x_{j_z}) \right\} + 2 \prod_{k=1}^{n} p_{10}(x_k) \tag{4}$$

where $p_{10}(x_i)$ and $p_{11}(x_i)$ are the probabilities of the signal $x_i$ to make the transitions $1 \rightarrow 0$ and $1 \rightarrow 1$ within two successive clock cycles, respectively. The factor 2 is arisen by the fact that $p_{10}(x_i) = p_{01}(x_i) = E(x_i)/2$ and thus, the derived combinations are multiplied by 2.

Eventually, the total power dissipation is given by:

$$P_{\text{tot}} = \frac{1}{2} V^2 f \left[ C_{\text{and}} \sum_{i=1}^{q} E(x_i) + C_{\text{or}} \sum_{j=1}^{m} E(y_j) \right] + P_{\text{out}} \tag{5}$$

where $x_i$ is the $i$-th input of the first level, $q$ is the total number of the inputs of the first level, $y_j$ is the $j$-th output of the first level, $m$ is the number of the AND gates, $C_{\text{and}}$ is the input capacitance of an AND gate, $C_{\text{or}}$ is the input capacitance of an OR gate and $P_{\text{out}}$ is the power dissipation at the external load.

## 3. SYNTHESIS FOR LOW POWER

### 3.1. Description of the New Idea

The basic principles of the proposed method employing a certain logic function are presented first, while the formal description follows. For the logic function $F = x_1x_2 + x_1x_3 + x_1x_4$, let us assume that the input signal $x_4$ exhibits low-transition probability and high static-1 probability and, the signals $x_1$, $x_2$, and $x_3$ are characterized by

high-transition probabilities. The logic circuit of the function $F$ is shown in Figure 1.

When $x_4 = 1$, the function can be expressed as $F_{x_4=1} = x_1x_2 + x_1x_3 + x_1 = x_1$. During this time interval the value of the logic function depends only on the value of the internal node $n_3$. More specifically; if $x_1 = 1$, the logic value of $F$ equals 1 for any logic value of the nodes $n_1$ and $n_2$. Hence, the corresponding power consumption of the nodes $n_1$ and $n_2$ can be reduced, *i.e.*, we can conserve energy. Using the complement form of the signal $x_4$ in the first two terms of the logic function, the new logic expression, $F' = x_1x_2\bar{x}_4 + x_1x_3\bar{x}_4 + x_1x_4$, is obtained, as it is shown in Figure 2. In this circuit, if $x_4 = 1$, the nodes $n_1$ and $n_2$ become zero and thus, they do not make any transition. The functionality of the original circuit is preserved since $F'_{x_4=1} = F'_{x_4=1} = x_1$ and $F_{x_4=0} = F'_{x_4=0} = x_1x_3 + x_1x_2$. Consequently, selecting the signal $\bar{x}_4$ as *blocking variable* (definition in Section 3.2), the logic operation remains unchanged but the nodes $n_1$ and $n_2$ dissipate less power compared with the starting form of the logic function $F$.

Using the specific function $F = \{9, 10, 11, 12, 13, 14, 15\}$ (Fig. 3) and applying the two-level optimization tool ESPRESSO, three groups of the minterms {12, 13,15, 14}, {13, 15, 9, 11}, and {15, 14, 11, 10} are obtained, as it is depicted in Karnaugh-map of Figure 3. Since we would like to exploit the statistic characteristics of the input signals, the complement signal, $\bar{x}_4$, must appear as many times as possible in the terms of the function $F$. It can be proved that the subsets of the

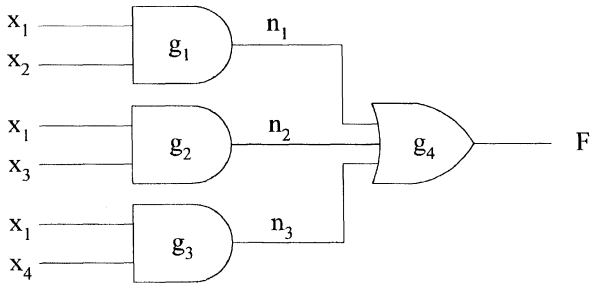



FIGURE 1 The logic circuit of function $F$.

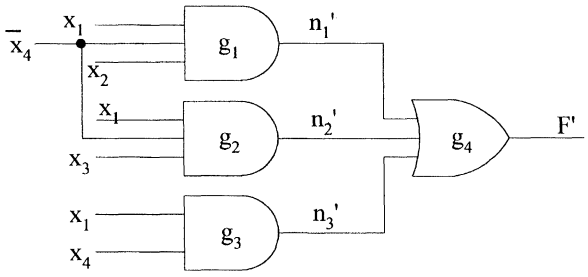
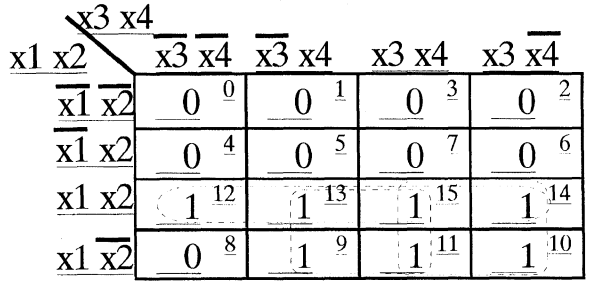
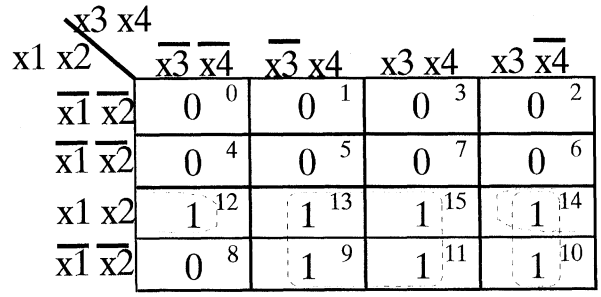



FIGURE 2 The Logic circuit of the modified function $F'$.




$F = x_1x_2 + x_1x_3 + x_1x_4$

FIGURE 3 Karnaugh map of the function $F$ for area optimization.




$F' = x_1x_2\bar{x}_4 + x_1x_3\bar{x}_4 + x_1x_4$

FIGURE 4 Karnaugh map of function $F'$ for power reduction.

minterms {9, 11, 13, 15}, {12, 14}, and {10, 14}, result into the power-optimized logic function $F' = x_1x_2\bar{x}_4 + x_1x_3\bar{x}_4 + x_1x_4$, as it is shown in Figure 4.

Let assume $P_1$ and $P_2$ be the total power dissipation of the two aforementioned logic functions $F$ and $F'$, respectively. Applying Eq. (5) for

each logic form, the corresponding total power dissipations are given by:

$$P_{\text{tot}1} = \frac{1}{2} V^2 f(C_{\text{and}}[3E(x_1) + E(x_2) + E(x_3) + E(x_4)] + C_{\text{or}}[E(n_1) + E(n_2) + E(n_3)]) \quad (6)$$

$$P_{\text{tot}2} = \frac{1}{2} V^2 f(C_{\text{and}}[3E(x_1) + E(x_2) + E(x_3) + 3E(x_4)] + C_{\text{or}}[E(n_1') + E(n_2') + E(n_3)]) \quad (7)$$

Assuming $p(x_1 = 1) = 0.5$, $E(x_1) = 0.7$, $p(x_2 = 1) = 0.5$, $E(x_2) = 0.8$, $p(x_3 = 1) = 0.4$, $E(x_3) = 0.6$, $p(x_4 = 1) = 0.9$, $E(x_4) = 0.2$ and using Eq. (5), it can be obtained that $E(n_1) = 0.47$, $E(n_2) = 0.37$, $E(n_3) = 0.66$, $E(n_1') = 0.05$, and $E(n_2') = 0.04$. Then, substituting the above values of the transition probabilities into Eqs. (6) and (7), the associated total power dissipations can be calculated by:

$$P_{\text{tot}1} = \frac{1}{2} V^2 f(C_{\text{and}}\, 3.7 + C_{\text{or}}\, 1.5) \quad (8)$$

$$P_{\text{tot}2} = \frac{1}{2} V^2 f(C_{\text{and}}\, 4.1 + C_{\text{or}}\, 0.75) \quad (9)$$

respectively. Hence, the expected total number of the transitions (over all nodes of the circuit) per clock cycle of the corresponding logic circuit are:

$$E_1 = 3.7 + 1.5 = 5.2 \quad (10)$$

$$E_2 = 4.1 + 0.75 = 4.85 \quad (11)$$

respectively. Apparently, the second logic implementation exhibits reduced transition activity compared with the first one. Eventually, using Eqs. (10) and (11), the resulting amount of the power saving is proportional to $((E_1 - E_2)/E_1) \cdot 100\% = 6.14\%$.

The above observation is the stimulus for the proposed method. The formal methodology for power optimization of a Boolean function expressed in minterms taking into account the statistic properties of the input signals will be presented in the next section. Exploiting the statistics of the input signals and using the main routines of ESPRESSO, a two level power minimization tool is developed.

### 3.2. Description of the Proposed Method

The goal of the introduced method is to reduce the switching activity of a two-level logic circuit. For that purpose, certain primary input signals with appropriate characteristics, as it has been presented in Section 3.1, must appear in the power-optimized logic expression. The selection of all logic variables (*i.e.*, blocking variables) with the appropriate statistic features is the first step of the proposed methodology. Then, these variables are used in the logic optimization process for reducing the switching activity as many as possible AND gates of a logic circuit. Also, it is possible to have input variables with identical or similar statistic features, which can be grouped into a number of sets, called *classes* hereafter. The fundamental principle that exists behind the class' notion is the partitioning of the Boolean space occupied by a logic function. Then, each partition (or class) will be optimized in terms of the power (*i.e.* blocking variables) and area employing the proposed method. Here, it should be stressed that area minimization is a critical issue in logic-level synthesis methods, since large logic circuits with respect to area lead to increased power dissipation [11, 12]. The final logic circuit results as the union of the area/power optimized partitions.

For the sake of clarity, an "informal" presentation of the basic principles of the introduced method has been done in the above paragraphs. Thus, before starting the detailed discussion of the proposed method, the precise definitions of the used principles have to be put forward.

DEFINITION 1 The set $C_l=\{x_i||E(x_i)-E(x_j)|\leq\varepsilon$ with $1\leq i,j\leq n,\varepsilon\in R_0^+\}$ is defined as a *class* of the input logic variables. The term $\varepsilon$ is a very small positive real number or zero specified by the designer.

The lower bound of classes number is equal to one, which implies that $|E(x_i)-E(x_j)|\leq\varepsilon\,\forall 1\leq i, j\leq n$, while the upper bound is equal to $n$, which means that $|E(x_i)-E(x_j)|\geq\varepsilon\ \forall\, 1\leq i,j\leq n$.

DEFINITION 2 Each member $x_i$ of the set $C_l$ is called *blocking variable*.

Let $C_1, C_2, \ldots, C_p$, $1\leq p\leq n$ be the different classes of the function $F$, taking into account the statistics properties of the primary inputs.

DEFINITION 3 *Transition Probability List* (TPL) is defined as the set

$$\begin{aligned}\text{TPL} = \{&C_1, C_2, \ldots, C_p | C_k \leq C_m \textit{ with } E(x_i)\\ &\langle E(x_j) \forall x_i \in C_k, \forall x_j \in C_m 1 \leq k, m \leq p,\\ &1 \leq p \leq n, \textit{and } 1 \leq i,j \leq n\}.\end{aligned}$$

Apparently, the above definition implies an ordered list of classes and provides a measure of the switching activity of each class. Also, the ordering specifies the "trajectory" of the optimization process through the logic variables.

DEFINITION 4 *Working Classes*, $k$, is defined as the number of classes, which are used for the partitioning of the boolean space.

All the above definitions are necessary for the blocking variables order. In other words, using the statistics of variables we select those variables which lead to optimal power synthesis.

A new synthesis tool for designing two-level logic circuits with minimal power dissipation is developed. The tool uses as input parameters the Boolean description $F=f(x_1, x_2, \ldots, x_n)$ (in Sum-Of-Products/Minterms), the static probabilities $P_F=\{p(x_1), p(x_2), \ldots, p(x_n)\}$, and the transition probabilities $E_F=\{E(x_1), E(x_2), \ldots, E(x_n)\}$ of the primary inputs.

The minterms expression is chosen as the starting logic form, since it is a "primitive" form. This implies that no transformations (for instance area minimization) have been applied on the logic function. Starting power optimization process from a different but logically-equivalent boolean form, a number of transformations would have been already applied.

**Step 1** The function $F$ is optimized by ESPRESSO tool and a new form $F'$ is derived. The total power dissipation of $F'$ and the average power dissipation per gate (*aver-*

The proposed synthesis procedure in algorithmic description is presented below:

**function** Power_Logic_Synthesis ($n$, $F$, *Stats*)

- $F$: Function to be optimized in Sum-of-Products (minterms).
- $n$: number of primary inputs, $x_i$, $i=1,2,\ldots,n$
- *Stats*: Statistics structure for each primary input contains:
- $p(x_i)$: Static-1 probability for input $x_i$.
- $E(x_i)$: Transition probability for input $x_i$.

**begin**

**Step 1** – ***ESPRESSO optimization and power calculation***

$F'=$espresso($F$); – $F'$ is the output of ESPRESSO tool.

*espresso_power* = power_estimation($F'$); – *average_espresso* is the average power per gate – of $F'$, *n_esp* is the number of cubes of $F'$

*average_espresso* = *espresso_power*/*n_esp*;

**Step 2** – ***Classes creation***

*TPL* = create_classes (*Stats*); – From input statistics file create the corresponding
– classes with $TPL = \{C_1, C_2, \ldots, C_p\}$, $1 \leq p \leq n$.

*SF* = create_sub_circuits(*F*, *TPL*, *p*); – Create $SF = \{F_1, F_2, \ldots, F_p\}$ sub-circuits from
– input *F*.

**Step 3** – ***Definition of power threshold and selection of "working" classes***

read (*AVERAGE*); – *power_threshold* is the power threshold of each cube.

*power_threshold* = *average_espresso* *(*AVERAGE*/100);

read(*k*); – Work with the first *k* classes of *TPL* where $1 \leq k \leq p$.

$SF' = \{F_1, F_2, \ldots, F_k\}$;

$F_{k+1} = SF - SF'$ – All the remaining minterms go to the class $F_{k+1}$.

**Step 4** – ***Optimization and power calculation in each working class***

**for** ($i = 1$; $i \leq k$; $i++$) – For each sub-circuit $F_1, F_2, \ldots, F_k$

{

$F'_i$ = espresso($F_i$); – Find the list of cubes $L_i = \{c_1, c_2, \ldots, c_{n_i}\}$ for $F'_i$.

**for** ($j = 1$; $j \leq n_i$; $j++$) – For each cube $c_1, c_2, \ldots, c_{n_i}$ of list $L_i$

{

**if** (power_of_cube($c_j$) ≥ *power_threshold*) **then**

{ – which dissipates more power than the *power_threshold*

delete_cube ($c_j$, *i*); – delete this cube $c_j$ from class $F'_i$.

*m* = find_proper_class($c_j$, *i*); – Find the "proper" class in which belongs to
– and insert it into the "proper" class

insert_cube_to_proper_class($c_j$, $F_m$); – $F_m$, $i + 1 \leq m \leq k + 1$

}–if

}–for *j*

}–for *i*

$F'_{k+1}$ = espresso($F_{k+1}$); – Run ESPRESSO for the sub-circuit $F_{k+1}$.

**Step 5** – ***Output circuit creation and power calculation***

$F'' = \{F'_1, F'_2, \ldots, F'_k, F'_{k+1}\}$

*our_ power* = power_estimation ($F''$);

**Step 6** – ***Re-definition of power threshold and working classes***

**if** (*our_power* ≥ *espresso_power*) **then** – Ask for new values of *AVERAGE* and *k*

**if** (prompt_for_new_iteration() = TRUE) **then**

goto **step 3**); – RUN steps 3 to 6 again.

**else** return ($F'$); – If there is no better power-solution, return $F'$.

**else** return ($F''$); – else there is a better power-solution, return $F''$.

**end** Power_Logic_Synthesis **function**;

*age_espresso*) is calculated using the power model proposed in Section 2.

**Step 2** Based on the input statistics, the *Transition Probability List* (TPL) is derived. Using this list the function *F* is split in *p* sub-functions $SF = \{F_1, F_2, \ldots, F_p\}$, where $p = |\text{TPL}|$. All minterms of each $F_i$ contain at least one blocking variable of the class $c_i$.

**Step 3** The *AVERAGE* coefficient (which determines the percentage of the *average_ espresso* mean power dissipation) and

the number $k$ of the working classes are specified by the designer. In addition, a set of $k$ sub-functions $SF' = \{F_1, F_2, \ldots, F_k\}$ is derived and the power threshold is calculated (based on the *AVERAGE* coefficient).

**Step 4** Power optimization of each sub-function $F_i$ results into the function $F_i'$ (function optimized in terms of area). The power consumption, $P_{c_j}$, of each cube $c_j$ of $F_i'$ is compared with the power threshold, *power_threshold*. In particular, if $P_{c_j} \leq$ *power_threshold* the cube $c_j$ remains in the current class $c_i$, while if $P_{c_j} >$ *power_threshold*, the cube $c_j$ is removed from the current class $C_i$ and reduced to two cubes. One of them is inserted in the class $C_m$, while the other in the class $C_n$, where $i+1 \leq m < n < k$. Cubes reduction aims at increasing the Boolean space that corresponds to the sub-functions $F_j$, where $i < j \leq k$.

**Step 5** The form $F''$ is obtained by the union of the $F_i'$s and thus, the total power dissipation is calculated.

**Step 6** Finally, in the user is prompted for new values of *AVERAGE* and $k$ if the resulted function $F''$ dissipates more power than $F'$.

## 4. RESULTS

The proposed method for minimizing power dissipation of a function $F$ is implemented by a new synthesis tool, which can be considered as a modified ESPRESSO. The tool takes as inputs the statistic information of the primary inputs and the Boolean expression of the function $F$.

In order to study the effectiveness of the proposed method a large number of logic functions have been optimized. More specifically, logic circuits of $n = 7, 8, \ldots, 10$ primary inputs have been chosen for optimization. For each $n$, three different on-set ratios $\{0.4, 0.7, 0.9\}$ and for each on-set ratio value five different circuits are considered. Furthermore, for each number $n$ of primary inputs, ten different randomly-distributed statistics files, which contain the static and transitions probabilities, have been taken into account.

For the calculation of the power dissipation it is assumed that: $C_{\text{or}} = 2\,C_g$ for OR inputs, $C_{\text{and}} = 1\,C_g$ for AND inputs, $C_{\text{INV}} = 1\ C_g$ for inverted inputs of an AND gate, and $(1/2)C_g V^2 f = 1$ (normalization).

Tables I and II show the experimental results for two different *AVERAGE* coefficients. Each table row (*i.e.*, specific $n$) represents 50 (*i.e.* $\{5$ circuits$\} \times \{10$ statistics files$\}$) generated logic

TABLE I Comparison results of ESPRESSO and the proposed method in terms of the first level outputs and total (normalized) power dissipation, with *AVERAGE* = 100 and $k = \lceil p/2 \rceil$

| # Primary inputs $n$ | On-set ratio (%) | First level outputs norm. power (ESPRESSO) | First level outputs norm. power (New method) | Difference (%) | Total norm. power (ESPRESSO) | Total norm. power (New method) | Difference (%) |
|---|---|---|---|---|---|---|---|
| 7 | 40 | 39,29 | 37,51 | −4,53 | 1498,26 | 1487,42 | −0.72 |
| 7 | 70 | 73,97 | 69,66 | −5,83 | 1325,54 | 1300,98 | −1,85 |
| 7 | 90 | 104,33 | 98,97 | −5,14 | 647,78 | 641,74 | −0,93 |
| 8 | 40 | 32,09 | 32,58 | +1,53 | 2501,73 | 2490,58 | −0,45 |
| 8 | 70 | 75,94 | 76,03 | +0,12 | 2007,73 | 2009,49 | +0,11 |
| 8 | 90 | 97,36 | 103,55 | +6,36 | 1080,36 | 1085,19 | +0,45 |
| 9 | 40 | 42,15 | 43,82 | +3,96 | 7065,77 | 6984,85 | −1,15 |
| 9 | 70 | 83,36 | 81,61 | −2,1 | 5433,57 | 5225,39 | −3,83 |
| 9 | 90 | 132,18 | 124,5 | −5,81 | 2546,58 | 2493,95 | −2.07 |
| 10 | 40 | 35,85 | 37,91 | +5,75 | 13664,65 | 13466,64 | −1,45 |
| 10 | 70 | 82,22 | 79,25 | −3,61 | 10462,96 | 10086,98 | −3,59 |
| 10 | 90 | 129,67 | 131,22 | +1,2 | 5055,97 | 4918,21 | −2,72 |

TABLE II Minimum Power Dissipation for $AVERAGE = 100$

| # Primary inputs $n$ | On-set ratio (%) | Min power (ESPRESSO) | Min power (New method) | Difference (%) |
|---|---|---|---|---|
| 7 | 40 | 70.05 | 65.72 | − 6.17 |
| 7 | 70 | 38.42 | 32.60 | − 15.15 |
| 7 | 90 | 24.24 | 19.15 | − 20.99 |
| 8 | 40 | 158.02 | 147.8 | − 6.47 |
| 8 | 70 | 100.25 | 88.15 | − 12.06 |
| 8 | 90 | 70.79 | 58.52 | − 17.47 |
| 9 | 40 | 385.23 | 367.42 | − 4.62 |
| 9 | 70 | 205.34 | 179.77 | − 12.46 |
| 9 | 90 | 109.98 | 93.86 | − 14.66 |
| 10 | 40 | 679.37 | 647.08 | − 4.75 |
| 10 | 70 | 378.03 | 344.10 | − 8.98 |
| 10 | 90 | 264.41 | 230.61 | − 12.78 |

circuits for each combination of $n$, on-set-ratio, and statistics. More specifically, Columns 3 and 6 give the normalized power dissipation of the outputs of the first level and the total power dissipation, respectively, when the ESPRESSO tool (*i.e.*, area optimization) is used. Columns 4 and 7 give the normalized power dissipation outputs of the first level and the total power dissipation derived from the proposed method.

From Table I it can be noticed that a power reduction of the total power dissipation can be achieved applying the proposed method. Moreover, the selection of the $AVERAGE = 100$ results into a "balanced" power dissipation and reduction between the total power dissipation and the power dissipation at the first level outputs. The Table II shows the maximum values of the power savings among the fifty (50) logic circuits of each row of Table I. It can be easily seen that the half number of the logic circuits exhibit power savings ranging from 12% and 20%, while the remaining circuits achieve gains ranging from 5% to 10%. Therefore, the obtained power savings imply that the proposed method can approaches the theoretical boundaries satisfactorily (for logic level design is about 20% [1, 2, 3]).

If the designer requires large power reduction only in the outputs of the first level, a large $AVERAGE$ (*e.g.*, $AVERAGE = 1000$) should be chosen. Table III indicates significant power savings in the nodes of the first level, while the total power consumption is increased comparing

TABLE III Comparison results of ESPRESSO and the proposed method in terms of the first level outputs and total (normalized) power dissipation, with $AVERAGE = 1000$ and $k = \lceil p/2 \rceil$

| # Primary inputs $n$ | On-set ratio (%) | First level outputs norm. power (ESPRESSO) | First level outputs norm. power (New method) | Difference (%) | Total norm. power (ESPRESSO) | Total norm. power (New method) | Difference(%) |
|---|---|---|---|---|---|---|---|
| 7 | 40 | 39,29 | 35,61 | − 9,37 | 1498,26 | 1690,37 | + 12,82 |
| 7 | 70 | 73,97 | 65,66 | − 11,23 | 1325,54 | 1610,48 | + 21,5 |
| 7 | 90 | 104,33 | 89,44 | − 14,27 | 647,78 | 885,18 | + 36,65 |
| 8 | 40 | 32,09 | 31,32 | + 2,4 | 2501,73 | 2761,52 | + 10,38 |
| 8 | 70 | 75,94 | 66,7 | − 12,17 | 2007,73 | 2206,35 | + 9,92 |
| 8 | 90 | 97,36 | 80,14 | − 17,69 | 1080,36 | 1335,61 | + 23,63 |
| 9 | 40 | 42,15 | 38,76 | − 8,04 | 7065,77 | 7834,76 | + 10,88 |
| 9 | 70 | 83,36 | 79,38 | − 4,77 | 5433,57 | 6394,15 | + 17,68 |
| 9 | 90 | 132,18 | 112,84 | − 14,63 | 2546,58 | 3249,08 | + 27,59 |
| 10 | 40 | 35,85 | 37,54 | 4,71 | 13664,65 | 14695,17 | + 7,54 |
| 10 | 70 | 82,22 | 76,51 | − 6,94 | 10462,96 | 11825,59 | + 13,02 |
| 10 | 90 | 129,67 | 116,37 | − 10,26 | 5055,97 | 5919,15 | + 17,07 |

with the associated power consumption of the area-optimized function. The increased power consumption comes from the fact that all cubes (*i.e.*, AND gate) remain in their class. Each cube has a lot of blocking variables resulting into an output with low activity. Although the choice of a large *AVERAGE* value results into increased power consumption, this can be used in multilevel circuits, which have nodes with large fanouts. Therefore, the reduction of the activity becomes extremely important factor of the total consumption, due to high capacitance of the node. It is apparent that the above design approach cannot be used as a two-level approach but as a technique in a multilevel design process.

## 5. CONCLUSIONS

In this paper, a new synthesis algorithm for minimizing the power dissipation of two-level logic functions is presented. Using the statistical properties of the primary inputs, appropriate variables have been selected for reducing power dissipation. The introduced synthesis method achieves remarkable reduction of the total power and/or the first level outputs power dissipation, depending on the selected *AVERAGE* and the number of working classes values (parameters). Future research work in this area will include the impact of using the blocking variables in a multilevel logic synthesis environment.

### *References*

[1] Chandrakasan, A., Sheng, S. and Brodersen, R. W. (1992). "Low Power Digital CMOS Design", In: *IEEE Journal of Solid State Circuits*, 473–484.

[2] Pedram, M. (1996). "Power Minimization in IC Design: Principles and Applications", *ACM Transactions on Design Automation of Electronic Systems* (*Todaes*), **1**(1), 3–56.

[3] Singh, D., Rabaey, J., Pedram, M., Catthoor, F., Rajgopal, S., Sehgal, N. and Mozdzen, T. (1995). "Power Conscious CAD Tools and Methodologies: A Perspective", In: *Proc. of IEEE*, **83**(4), 570–594.

[4] Brayton, R. K., Hachtel, G. D., McMullen, C. T. and Sangiovanni-Vincentelli, A. L. (1984). "*Logic Minimization Algorithms for VLSI Synthesis*", Kluwer Academic Publishers.

[5] De Micheli, G. (1994). "*Synthesis and Optimization of Digital Circuits*", Mc Graw Hill.

[6] Devadas, S., Gosh, A. and Keutzer, K. (1994). "*Logic Synthesis*", Mc Graw Hill.

[7] Rudell, R. L. and Sangiovanni-Vincentelli, A. L. (1987). "Multiple-valued minimization for PLA optimization", In: *IEEE Transactions on Computer-Aided Design*, **CAD-6**(5), 727–750.

[8] Brayton, R. K., McGeer, P. C., Sanghavi, J. V. and Sangiovanni-Vincentelli, A. L. (1993). "A New Exact Minimizer for Two Level Logic Synthesis", In: *Logic Synthesis and Optimization*, Chapter 1, pp. 1–31, Kluwer Academic Publishers.

[9] Shen, A. A., Ghosh, A., Devadas, S. and Keutzer, K., "On average power dissipation and random pattern testability of combinational logic networks", In: *Proc. of the Int. Conf. on Computer Aided Design*, November 1992, pp. 402–407.

[10] Vrudhula, S. B. K. and Xie, H. Y. (1994). "Techniques for CMOS Estimation and Logic Synthesis for Low Power", In: *Proc. of Int. Workshop on Low Power Design*, Napa Valley, pp. 21–26.

[11] Iman, S. and Pedram, M., "Two-Level Logic Minimization for Low Power", In: *Proc. of Int. Conf. on Computer Aided-Design* (*ICCAD*), Nov. 1995, pp. 433–438.

[12] Iman, S., "Logic Synthesis for Low Power VLSI Designs", *Ph.D. Dissertation*, Univ. of Southern California, August 1996.

[13] Schneider, H. and Schlichtmann, U. (1994). "Decomposition of Boolean Functions for Low Power Based on a New Power Estimation Technique", In: *Proc. of Int. Workshop on Low Power Design*, Napa Valley, pp. 123–128.

### 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.

**Dr. 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 Lecturer 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 30 papers in international journals and conferences. He is a member of the IEEE and ACM.

**Dr. 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 110 papers in international journals and conferences. He has been awarded a large number of Research Contracts from ESPRIT, RACE, and National Programs.