IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 June 2024, accepted 5 August 2024, date of publication 13 August 2024, date of current version 23 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3442975

RESEARCH ARTICLE

Switching Activity Reduction of SOP Networks

MARCIN KUBICA[1], BOLESŁAW POCHOPIEŃ[2], AND DARIUSZ KANIA[1]
[1]Department of Digital Systems, Silesian University of Technology, 44-100 Gliwice, Poland
[2]Department of Graphics, Computer Vision and Digital Systems, Silesian University of Technology, 44-100 Gliwice, Poland
Corresponding author: Marcin Kubica (mkubica@polsl.pl)

ABSTRACT An extremely important aspect of the synthesis of digital circuits is obtaining solutions optimized in terms of energy consumption. This can be achieved by limiting the dynamic power, depending on the switching activity of the nodes of the implemented logic network. The article proposes a new technology mapping method leading to a reduction in switching activity. An approach known from the literature using an output graph was improved, the essence of which lies in the analysis of the created model of the switching system. The essence of this improvement was to take into account additional parameters related to switching the gateway network. The main contribution of the article is the use of a switching model in the form of an innovative switching graph to limit the number of switches of the logic network. The presented approach is focused on the sum-of-products (SOP) architecture. The reason for this choice is that it is the most elementary and universal form of mapping a combinational function, often implemented in some families of programmable devices. The proposed approach resulted in a significant reduction in switching activity compared to the classical implementation, by 68.9%, 70.6% and 70.2%, respectively, for SOP architectures with 3, 5 and 7 input sums. Of the 20 test systems considered, no significant improvement in switching activity was achieved compared to the classical method, only for 2, 3 and 4 cases, respectively, for SOP architectures with 3, 5 and 7 input sums. The results presented in the article confirm the effectiveness of the proposed methods.

INDEX TERMS Low power synthesis, SOP, switching activity, technology mapping.

I. INTRODUCTION

The significant increase in the complexity of digital systems still raises questions about effective methods of designing these systems. Naturally, the effectiveness can be measured by various factors, such as: speed of operation, reliability, the number of logic resources needed for implementation or their cost. However, it turns out that in many applications the key factor is the power consumption of the implemented system. Energy efficiency is extremely important in areas such as military, medical and space applications. Most often, stringent energy efficiency requirements are associated with the need to use battery power. This is usually related to the mobility of this type of devices. It should also be emphasized that the energy efficiency of specific systems also affects their other properties. Typically, energy-efficient systems are also more reliable because the amount of heat energy needed to be dissipated is lower, reducing the likelihood of excess temperature damaging the logic structure. It is worth adding that limiting power consumption is usually also associated with limiting the complexity of the logic structure, and thus limiting the number of logic resources used. In this situation, we can often talk about cost reduction. All these observations clearly show that limiting the energy consumed by a digital system is an extremely important aspect of logic synthesis.

The process of reducing the energy consumed can be e.g. appropriate partition of tasks into software and hardware parts [1], [2], [3], [4], or temporarily turning off selected modules. Most often, in the case of high-level synthesis, we are talking about reducing energy consumption at the system level, as shown, among others, in the works [2], [4], [5], [6]. It should be emphasized that the reduction of energy consumption depends on the specificity of the designed system [7], [8], [9]. There are many papers in the literature discussing this problem [5], [10], [11], [12], [13], [14].

The associate editor coordinating the review of this manuscript and approving it for publication was Stavros Souravlas.

Naturally, it is possible to reduce energy consumption also at the low-level synthesis stages. More important methods of reducing power consumption include the idea of local reduction of the supply voltage [15], [16], [17], [18] or the ''power gating'' method [14], [19], [20]. For synchronous circuits, good results can be obtained by limiting the frequency of the clock signal. Of course, this approach is unfavourable from the point of view of the speed of operation of a given circuit. In this situation, a better solution is to temporarily gate the clock signal (clock gating) [21], [22], [23], [24], [25], [26], or only locally lower the frequency of the clock signal in selected modules of the designed system [13], [27], [28]. Another method of reducing energy consumption is to implement circuits in the form of a GALS (Globally Asynchronous Locally Synchronous) structure [29], [30], [31], [32], [33]. This concept allows only selected parts of the circuit to be implemented as synchronous, while data exchange within the entire system usually takes place asynchronously [34]. Naturally, there are many more low-level concepts for reducing energy consumption such as: appropriate scheduling method [9], reducing the logic levels in digital hardware [10], using a Pipelined Vedic Multiplier [35], appropriate instruction encoding [36], based on the concept of computational kernel [37], improving characteristics of FSMs [38], [39]. They are often targeted at specific systemic applications [40], [41], [42], [43]. In the case of implementing an FSM, excellent results are achieved by appropriate coding of the internal states of the FSM (dynamic power reduction of 8%) [44], [45], [46], [47], [48], [49], [50]. Sometimes a synthesis of low power FSM includes elements of artificial intelligence [51]. However, in the case of combinational circuits, combining the problem of energy consumption reduction with function decomposition leads to good results (reduction of switching activity from 6 to 40%) [52], [53], [54], [55]. In general, reducing energy consumption is often based on reducing the number of switchings of the implemented circuit, which is the main topic of this paper.

The paper focuses on the idea of minimizing the number of switchings of combinational circuits with sum-of-products architecture. The SOP architecture is a representation of the classical form of describing a logical function associated with a set of implicants. The description in the form of a set of implicants has a universal character. It is also represented in hardware form in the form of CPLD (Complex Programmable Logic Device) devices. The representation of a function in the form of a set of implicants is a starting point for other, more complex factorization oriented to multi-level implementation. This approach can be the basis for reducing the energy consumed by combination blocks included in the digital system.

Over the years, many methods for optimizing combinational circuits have been developed. Initially, the main emphasis was on the search for two-level optimization. This led to the development of optimal two-level minimization methods, which had a direct impact on minimizing the number of products. The effect of this method was to minimize the number of gates needed to implement logic expressions, which in the case of CPLD systems led to the minimization of the silicon surface. Later, two-level minimization became the basis for multi-level minimization, primarily aimed at minimizing the logical resources used, such as AND-OR gates or LUT blocks contained in FPGA structures. One of the concepts of multi-level optimization, the starting point of which is two-level minimization, is our function mapping method using output graphs [59], [60], [61]. In the first works, they were used to search for an implementation optimized in terms of area, which was expressed by the number of the so-called PAL blocks [59], [60]. It was also possible to search for solutions containing a minimized number of logic levels [61], [63]. This method was also extended to the synthesis of sequential circuits, in which optimization consisted of using the Graph of Excitations and Outputs. Currently, the most important aspect of synthesis is the minimization of energy consumption, which is why it was decided to use previous achievements in the synthesis process focused on power minimization. The problem of power minimization has been the subject of many works aimed at appropriate coding of FSM states [46], [47], [48], implementation of circuits in the form of GALS structures [34], or the use of original architectures of specialized systems [12]. The experience gained allowed us to develop further improvements in digital systems in terms of power consumption. The idea of minimizing switching of combinational circuits implemented in the form of SOP networks is a link to the concept of mapping circuits described with an output graph in the form of PAL blocks commonly found in CPLD systems. The concept of describing circuit switching in the form of a switching graph presented in the article allows for the optimization of the combinational part of the circuits in terms of the number of switchings, and thus dynamic power losses.

The article presents an output graph known from the literature, which was adapted to obtain a switching model. The essence of this improvement was to include additional parameters related to switching SOP networks.

The main contribution of the article is the use of a switching model in the form of an innovative switching graph to limit the number of switches of the logic network. This led to the creation of a new technology mapping model that focused on limiting switching activity.

The second section discusses the basic concepts related to static and dynamic power and switching activity. Sections III and IV discuss the classical implementation of logic functions and the implementation based on graphs of outputs, respectively. Implementation in the form of SOP architecture is considered. Sections V presents an innovative approach to the implementation of a multi-level system, leading to the reduction of switching activity in the process of technology mapping. A new form of describing the energy properties of technology mapping in the form of a switching graph was proposed. Section VI proposes a methodology for using the

graph of outputs in the technology mapping process, enabling the observation of the energy properties of the obtained solution. The last two sections contain the experimental results and conclusions, respectively.

II. THEORETICAL BACKGROUND

The power consumed by a digital circuit can be divided into static power and dynamic power. Static power is lost in a digital circuit that is in a stable state, i.e. one in which there are no changes at the inputs and outputs of individual gates. Dynamic power refers to power losses occurring in transient states and is the result of energy losses occurring during switching of digital circuit elements.

Static power depends mainly on the values of subthreshold currents, drain leakage currents and gate leakage currents, which strictly depend on the technology in which the integrated circuit was made. If you want to implement an energy-saving circuit, you should use devices made with technology that guarantees the lowest possible static power losses.

The second power component is dynamic power, which depends not only on the circuit technology. It can be minimized by an appropriately conducted logic synthesis process. It results from the need to recharge the load capacitance of individual nodes.

The power necessary to charge or discharge the load capacity can be expressed as the relationship (1) [56]:

$$P_d = \left(\frac{1}{2}p \cdot C_L \cdot V_{dd}^2 \cdot f_{clk}\right) + (t_{sc} \cdot V_{dd} \cdot I_{sc} \cdot f_{clk}) \quad (1)$$

where p is the probability of state change, C_L – load capacitance, f_{clk} clock signal frequency, V_{dd} – supply voltage, t_{sc} – time during which both transistors (NMOS and PMOS) are turned on and I_{sc} – current in the stage of output gate.

The first component represents the power needed to switch the gate. The second component represents the power associated with the simultaneous conduction of the output transistors. This is due to the fact that during the output state change for a short time (t_{sc}), both the NMOS and PMOS transistors are conducting. This causes current (I_{sc}) to flow in the stage of output gate. Since the rise and fall times of signals in modern technologies are very short, the main component of dynamic power is the power associated with gate switching. This means that the dynamic power can be described with a very good approximation by the formula (2):

$$P_d = \frac{1}{2}p \cdot C_L \cdot V_{dd}^2 \cdot f_{clk} \quad (2)$$

The probability of switching the i-th output of the gate is equal to a coefficient called switching activity (SW_i). The SW_i value is defined as the number of changes in the logic state at the output of the i-th gate (n_i) in time T. In the case of aperiodic signals, they can be associated with the relation (3).

$$SW_i = \lim_{T\to\infty} \frac{n_i}{T} \quad (3)$$

The signal at the outputs of individual gates is usually not periodic, therefore, in the relationship determining the dynamic power, instead of the frequency f_{clk}, time T is used. Switching activity can be determined for each node of the system. It determines how often the logic signal changes in individual nodes of the circuit. Knowing the number of nodes in the circuit (n), the dynamic power related to the overload of the load capacity can be determined according to the relationship (4)

$$P_d = \sum_{i=1}^{n} \frac{1}{2T} SW_i \cdot C_{L_i} \cdot V_{dd}^2 \quad (4)$$

where SW_i means the switching activity of the i-th node of the circuit, and C_{Li}- the load capacity occurring in the i-th node.

With some simplification, it can be said that the dynamic power consumed by the circuit is proportional to the square of the supply voltage, average load capacitance of nodes, and switching activity. The first two parameters are closely related to the device's manufacturing technology, while the total switching activity is a parameter that we can influence in the synthesis process. Therefore, when optimizing dynamic power losses in the logic synthesis process, care should be taken to ensure that they change the logic states in individual circuit nodes as rarely as possible during operation. This can be achieved, for example, by appropriate coding of the internal states of the FSM or by designing a combinational circuit ensuring that signals associated with greater switching activity are located in the combinational network as close to the outputs as possible. The above observation is the essence of the idea presented in the paper, which boils down to the multi-level implementation of combination blocks whose total value of switching activity is lower than in the classical cascade implementation, without sharing logical resources.

Knowing the values of the probabilities of the value 1 appearing for individual inputs *p(x),* it becomes possible to determine the probability of the value 1 appearing at the outputs of the gates *p(y).* Examples of probability values for NOT, AND and OR gates are shown in Fig. 1.

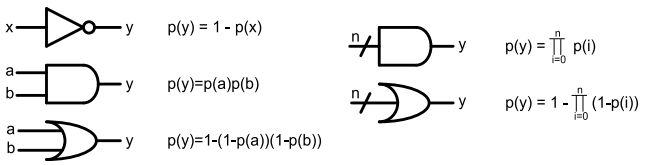

FIGURE 1. Probabilities of the value 1 appearing at the output of selected logic gates.

Knowing the probability of the value 1 appearing in a given node, which we assume is equal to *p(x)*, it becomes possible to determine the switching activity in accordance with the relationship (5).

$$SW = 2p(x)(1\text{-}p(x)) \quad (5)$$

In expression (5), the part (1-p(x)) describes the probability of the value 0 appearing in a given node. Thus, the product

p(x)(1-p(x)) describes the probability of transition from 1->0, and the product (1-p(x))p(x) from 0->1. Out of the four possible transitions (0->0, 0->1, 1->0, and 1->1), only two cause the node to switch to the opposite state, so the described product should be multiplied by 2 to obtain expression (5).

III. CLASSICAL IMPLEMENTATION OF COMBINATIONAL CIRCUITS IN THE SUM-OF-PRODUCT FORM

A logic function can be represented in many different forms. One of the more common forms is Sum-Of-Product form (SOP). The essence of this form is the presence of a single OR gate realizing the sum of components (products), which are factors of the logic value 1. It should be emphasized that in real implementations there are certain technological limitations. The number of inputs to individual gates is limited. This is especially important in the case of an OR gate. Let us denote the number of products connected to the OR gate as k. Therefore, knowing the form of the logic structure and the limitations on the number of terms connected to the OR gate (k), we can talk about the implementation of the function in the form of an SOP-type architecture.

In practical applications, the multi-output function is much more common than a single function. Naturally, the SOP form can also be used to realize a multi-output function. A good example of describing a multi-output function using the SOP form is the.pla format, where appropriate output vectors are associated with the appropriate input vectors (corresponding to the SOP implementation) [57].

An extremely important element of logic synthesis is the issue of technology mapping. So let's consider a situation in which a multi-output function is implemented rather than a single-output function. It turns out that the multi-output function mapping is very often implemented in a classical way [58]. The basis of this method is the implementation of each single-output function included in the multi-output function separately. There is no sharing of logical resources. A cascade structure is created for each individual single-output function, which increases the number of available products by creating subsequent logical levels. This process continues until this number is sufficient to perform the considered single-output function.

So let's consider the implementation of an example function f: B^4 -> B^3, where B = {0,1} in the SOP form. The description of the function in the form of a pla file, after two-level minimization, is shown in Fig. 2.

Let us assume that the technology mapping of the considered multi-output function will be implemented in the SOP architecture in which the value of k = 3 (three-input OR gates). To simplify the analysis, restrictions on the number of inputs to AND gates have been abandoned. The resulting gate structure is shown in Fig 3.

Analyzing the solution in Figure 3, it can be seen that three separate logic structures have been obtained for individual functions. In the case of functions f1 and f0, the obtained structures have a cascade character, because in both cases the number of required terms (5) is greater than the value of k.

Inputs	Outputs
a b c d	f2 f1 f0
1111	100
0100	100
0010	100
0001	010
111-	010
1000	010
0100	010
0010	010
1111	001
1000	001
0001	001
01-0	001
0-10	001

```
.i 4
.o 3
.ilb a b c d
.ob f2 f1 f0
.p 13
1111 100
0100 100
0010 100
0001 010
111- 010
1000 010
0100 010
0010 010
1111 001
1000 001
0001 001
01-0 001
0-10 001
.e
```

FIGURE 2. Description of an example function f: B^4 -> B^3.

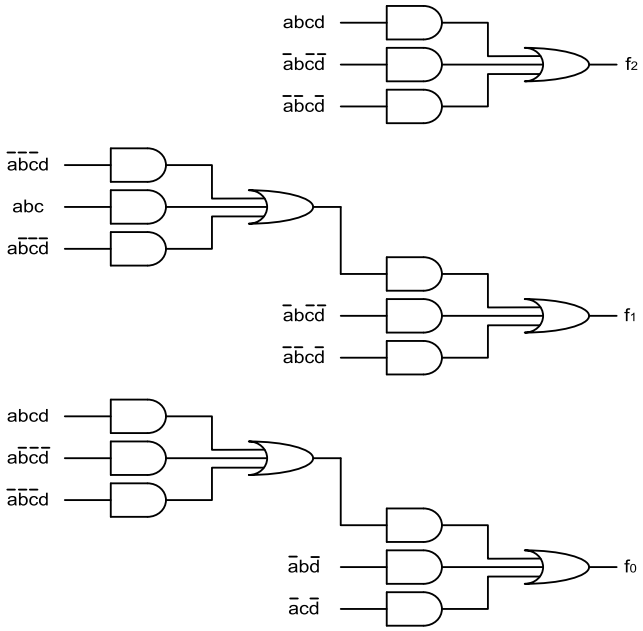

FIGURE 3. Technology mapping of the function f: B^4 -> B^3 using the classical method.

It is clearly visible that the classical method is ineffective both in terms of the number of necessary logic resources and the dynamic properties of the obtained solutions. Therefore, it is necessary to propose another method of technology mapping leading to the sharing of some logical resources between the individual single-output functions.

IV. IMPLEMENTATION OF COMBINATIONAL CIRCUITS USING AN OUTPUT GRAPH

The minimized form of the function (minimization of multi-output function) f: B^n -> B^m, where B = {0,1} can be associated with an graph of outputs whose nodes correspond to the output parts of multi-output implicants [59]. Let y_i be an m-element output vector corresponding to the output part of the i-th multi-output implicant. Let us call the discriminant Δy_i a decimal number equal to the number of vectors y_i appearing in the set of multi-output implicants describing the

function f. Let $\mu(\Delta y_i)$ be a decimal number equal to the number of elements $\{1\}$ contained in the vector y_i. The graph of outputs G(Y, U) is a directed graph, where Y is the set of all nodes Δy_i, and the set of edges U connecting nodes Δy_{si}, Δy_{ri} such that the code distance of the vectors y_{si}, y_{ri} is 1 and $\mu(\Delta y_{si}) + 1 = \mu(\Delta y_{ri})$. The essence of the output graph is presented in Figures 4a and 4b.

The general form of the graph of outputs for the set of three functions is shown in Figure 4a. Let us consider the multi-output function *f:* B^4 -> B^3 from Figure 2. This example, subjected to multi-output function minimization, takes the form as in Figure 4b. Therefore, it is possible to create an original graph of outputs for the considered example, as shown in Figure 4c. By eliminating the nodes for which $\Delta y_i = 0$ from the original graph, we obtain the reduced graph of outputs shown in Figure 4d.

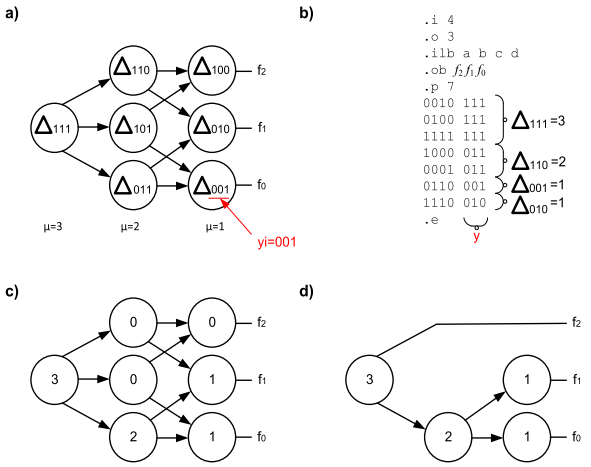

FIGURE 4. Relationship between the multi-output function description and the output graph: general form of output graph for a set of three functions (a), description of pla after minimizing the multi-output function *f*: B^4 -> B^3 (b), original output graph for a multi-output function *f*: B^4 -> B^3 (c), reduced graph outputs for multi-output function *f*: B^4 -> B^3 (d).

Based on the reduced graph of outputs, technology mapping can be performed. Between individual graph nodes, which correspond to single architecture-dependent logic structures, there are edges that represent connections between these structures. The resulting logic network is characterized by the sharing of some logical resources. In the example under consideration, we aim to map the multi-output function in the SOP architecture, which is why the nodes are mapped in structures composed of k-input OR gates with attached terms. For the graph shown in Figure 4d, the mapping shown in Figure 5 was obtained.

Ensuring resource sharing through the implementation of technology mapping using graph of outputs leads to a reduction in the used logical resources. It turns out that graph of outputs can also help in other types of technology mapping optimization. Various strategies for using graph of outputs in the process of technology mapping of logic functions aimed at optimizing the area or speed of the system, supported by a number of in-depth theoretical analyzes and examples, can be found in the papers [60], [61]. It is also possible to use the generalized form of graph of outputs, i.e. graph of excitations and outputs, in the process of optimizing the technology mapping of FSM in CPLD structures [62], [63]. The question arises whether graph of outputs can also be used to reduce the switching activity of the obtained structures, which may lead to a reduction in dynamic power.

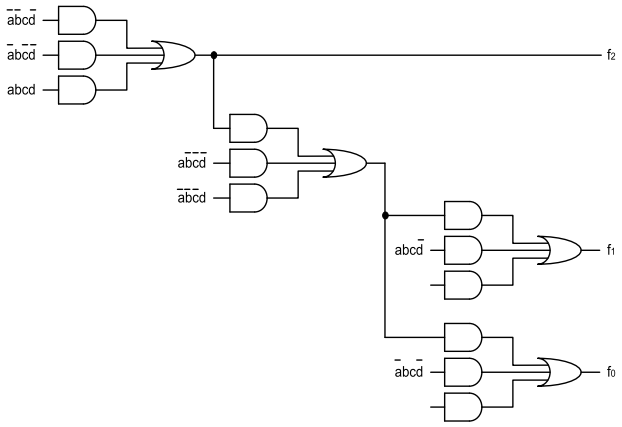

FIGURE 5. Technology mapping of the function *f*: B^4 -> B^3 using the graph of outputs.

V. TECHNOLOGY MAPPING TAKING INTO ACCOUNT THE SWITCHING MODEL

The technology mapping models presented earlier can be easily extended to include information about both the probability of occurrence of 1 at selected nodes and the SW values associated with individual SOP modules. Such a model will be called the switching model.

A. SWITCHING MODEL FOR CLASSICAL IMPLEMENTATION

Knowing the expressions implemented by individual terms, it becomes possible to determine the switching activity of the SOP. Assuming that the input variables (a, b, c, d) are independent and that the value 1 appears on each input with a probability of 0.5, it is possible to determine the probability of the value 1 appearing at the outputs of the AND gates and the output of the OR gate.

Consider the function f2, which takes the value 1 for three input vectors (1111, 0100, 0010). The probability values are determined according to the relationships (6) and (7).

$$\begin{aligned} p(abcd) &= p\left(\bar{a}b\bar{c}\bar{d}\right) = p\left(\bar{a}\bar{b}c\bar{d}\right) \\ &= p(a)\,p(b)\,p(c)\,p(d) = 0.5^4 = 0.0625 \end{aligned} \tag{6}$$

$$\begin{aligned} p(f_2) &= 1 - (1 - p(abcd))\left(1 - p\left(\bar{a}b\bar{c}\bar{d}\right)\right)\left(1 - p\left(\bar{a}\bar{b}c\bar{d}\right)\right) \\ &= 1 - 0.9375^3 = 0.176025 \end{aligned} \tag{7}$$

Knowing the probability of the value 1 appearing on all terms and the SOP output used to implement f2, it becomes necessary to determine the total value of switching activity

SW(f2)related to the implementation of function f2, which is 0.642 (8).

$$SW(f_2) = 3 * (2 * 0,0625 * (1 - 0,0625)) + 2 * 0,176025 * (1 - 0,176025) = 3 * 0,117188 + 0,290081 = 0,641643 \cong 0,642 \quad (8)$$

To implement the f1 and f0 functions, a larger number of terms is needed, which makes it necessary to cascade them, assuming the use of SOP blocks with 3 terms. From the point of view of power consumption, it is enough to associate each SOP with the total value of switching activity. It is also important to determine the probability of the value 1 appearing at the input of each SOP, which allows determining the power consumption of the entire circuit of which individual SOPs are a component. To do this, it is necessary to analyze the network successively, starting from the SOPs located closest to the inputs and ending with the network elements generating output signals. In this situation, it becomes possible to present a switching model that describes the energy properties of the classical implementation of the exemplary function f: $B^4 \rightarrow B^3$ in the form of an SOP network (Fig. 6).

The proposed switching model was described in such a way that the nodes contained values describing the total value of the switching activity coefficient, and the outputs contained the probability of the value 1. The numbers next to the arrows entering the node indicate the number of literals included in the individual terms.

The switching model can be related directly to the obtained logic structure by assigning parameter values resulting from the model at appropriate points in the logic network, as shown in Figure 7.

B. SWITCHING MODEL FOR IMPLEMENTATION USING GRAPH OF OUTPUTS

It turns out that graph of outputs can be perfectly adapted to describe the switching properties of technology mapping. We obtain the switching model of technology mapping by adding switching parameters to individual nodes of the graph of outputs. Such a graph will be called a switching graph.

Each node of the switching graph is associated with a group of implicants that can be implemented in the form of an SOP. The edges connecting the nodes of this graph represent the terms of the SOP located at the next level. For each node, you can determine the switching activity coefficient and the probability of the value 1 appearing at the output. Assuming the probability of the value of 1 for individual input variables equal to 0.5 and starting the analysis from the nodes of the highest-level graph, it becomes possible to determine the parameters for each node and determine all the parameters of the switching model.

The switching graph for the set of functions f: B^4->B^3 is presented in Figure 8.

The switching graph can, of course, be directly linked to the obtained SOP network by assigning parameter values

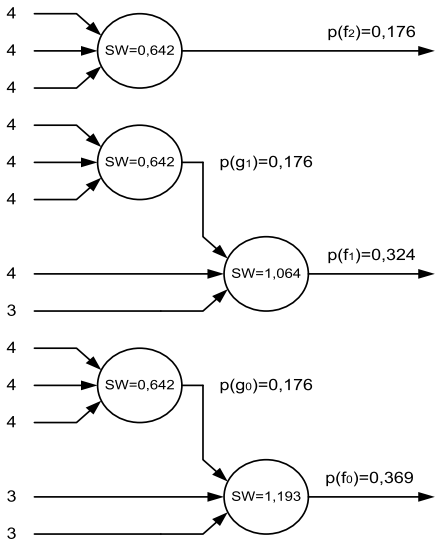

FIGURE 6. The switching model of a classical implementation for an example function f: B^4 -> B^3.

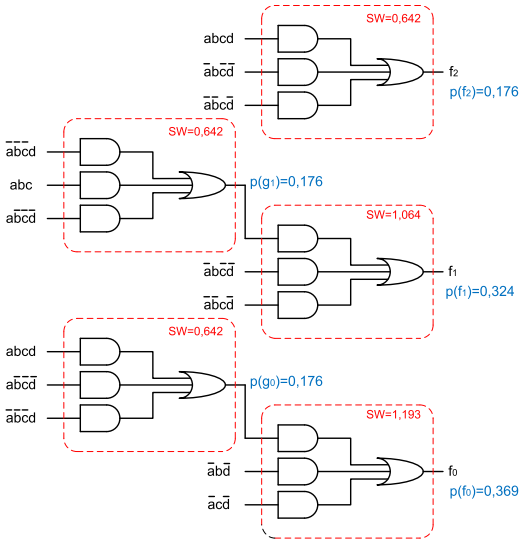

FIGURE 7. Technology mapping of the function f: B^4 -> B^3 using the classical method, along with switching parameters.

resulting from the graph to appropriate points in the logic structure, as shown in Figure 9.

By summing up the values of switching activities occurring in the individual cases presented in Fig. 7 and Fig. 9, we can compare the considered implementations in terms of the dynamic power used. In the case of the Classical implementation, the switching model of which is presented in Fig. 7, the value of $SW_{cl} = 4.182$. In the implementation using the switching graph, $SW_{sg} = 3.471$. In this situation, the multi-level implementation uses SW_{sg}/SW_{cl} *100% = 0.83% of the dynamic power of the Classical implementation.

VI. METHODOLOGY FOR IMPLEMENTING TECHNOLOGY MAPPING USING A SWITCHING GRAPH

The considerations presented in the previous section make it possible to propose the following methodology for

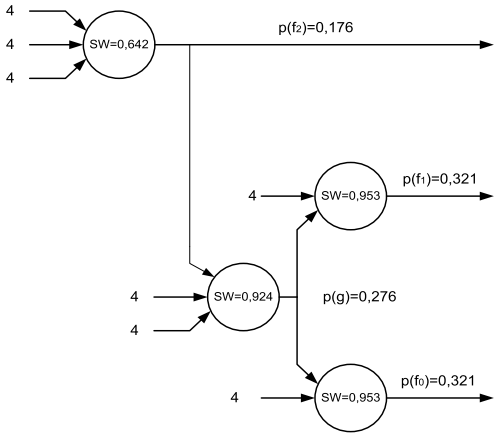

FIGURE 8. Switching graph for function f: B^4 -> B^3.

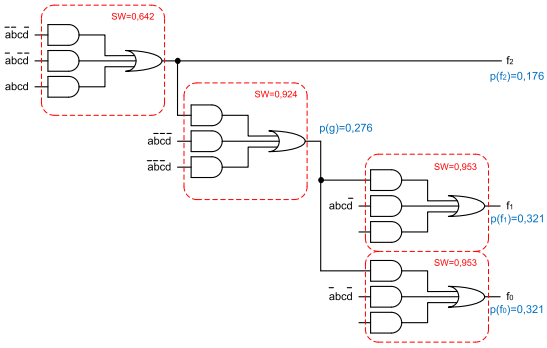

FIGURE 9. Technology mapping of the function f: B^4 -> B^3 using a switching graph.

technology mapping of the multi-output function aimed at minimizing the number of switches. The presented technology mapping methodology was used to determine the coefficients necessary to determine the dynamic power.

The methodology consists of the following steps:

1. Minimizing multi-output function

2. Based on the output vectors obtained after minimization, determine the values of the discriminants Δy_i and their orders $\mu(\Delta yi)$.

3. Creation of the primary graph of outputs.

4. Reduction of the primary graph of output to a form in which only nodes remain for which $\Delta y_i \neq 0$.

5. Implementation of multi-output implicants associated with the selected node of the graph of outputs in the form of SOPs according to the following rules:

- we choose a node for which $\mu(\Delta\ y_i) = \max$
- if $\Delta y_i > k$, we implement the implicants related to the selected node using the classical method

6. For the technology mapping of the graph of outputs node selected in step 5, we determine the total SW value and the probability p(out), where p(out) is the probability of the value 1 appearing at the output of the OR gate belonging to the SOP block.

7. Creation of a node switching graph describing the technology mapping of the graph of outputs in the form of a network of SOP blocks, reduction of the graph of outputs by eliminating the node that has been mapped and go to 5.

8. If all nodes of the graph of outputs have already been mapped, based on the switching graph we determine the total value of the SW coefficient for the obtained mapping in the form of an SOP network.

9. We determine the coefficient showing the dynamic power savings obtained for the technology mapping compared to the classical implementation.

The described methodology is also presented in Figure 10.

VII. EXPERIMENTAL RESULTS

In order to confirm the effectiveness of the proposed solutions, a number of experiments were carried out. The experiments were performed through mathematical analysis using a specially prepared spreadsheet. The popular benchmark set [64] was used for experiments. In most benchmarks, only a subset of functions f was selected: f: B^n -> B^m, such that the number of m was limited to the value 3 or 4, so that ''manual'' analysis of selected examples in the spreadsheet of the proposed methodology was possible. Two spreadsheets were prepared for each case: for the classical method (Cl) and for the method using the graph of outputs, transformed into an appropriately created switching graph (SG). The aim of the experiments was to determine the total SW value for both presented implementations. The experiments were carried out for three values of k (3, 5 and 7) defining the number of terms in SOPs.

The results are summarized in Table 1. The first three columns with the ''Benchmark'' heading contain information about the benchmarks used in the experiments, i.e. the name of the benchmark (Name), the number of inputs (in) and the number of outputs (out). The name of the example used (Name column) contains information about what subset of outputs was selected from the original benchmark. If the name of the benchmark does not contain markings indicating specific outputs, the benchmark covering all outputs was analyzed. In some cases, the number of inputs given in the (in) column is less than the number of inputs for the original benchmark. This means that for the functions considered there are insignificant variables that have been omitted. The value contained in the column with the ''out'' header directly determines the number of functions constituting the subassembly that was analyzed in the considered case. The rest of the table is divided into three parts depending on the form of the SOP architecture. The considerations are limited to the cases of OR gates with 3, 5 and 7 inputs. In each of the 3 cases, the SW value was determined for the structure determined using the classical method (Cl) and for the structure described using a switching graph (SG). Additionally, Table 1 includes a column presenting a comparison of the total values of SW coefficients obtained for the presented solutions as Cl/SG. The values there indicate how many times the total value of the SW coefficient is greater for the classical implementation (CL) in relation to the implementation obtained on the basis of the graph of outputs, and from the point of the number

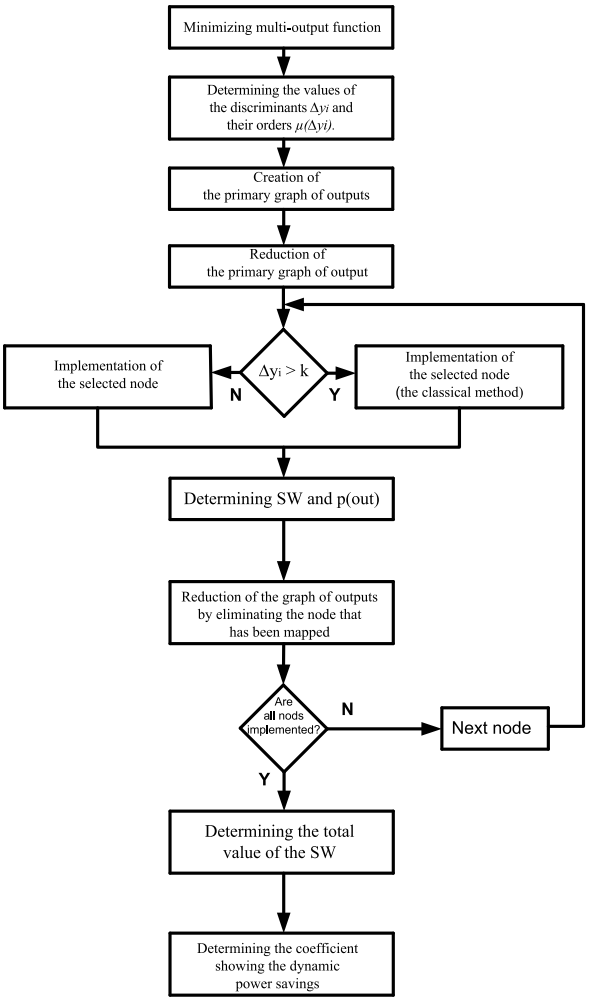

FIGURE 10. Methodology for implementing technology mapping using a switching graph.

of switchings described in the form of a switching graph (SG). CL/SG ratio values were determined for individual benchmarks. According to relation (4), the obtained CL/SG coefficients can be directly related to the values of dynamic power needed to switch SOP networks.

Additionally, the penultimate row of the table (SUM) contains the total values of the SW coefficients obtained for the Cl and SG implementations, for each of the three values of the k parameter. A good indicator for comparing the proposed methods is the average value of the CL/SG coefficients obtained for individual benchmarks, included for each part of the table in the last row with the MEAN description. It indicates average benefits related to the reduction of the number of switches for the analyzed set of benchmarks.

The obtained coefficient values included in the last two rows of the table are presented in the form of bar graphs presented in Figure 11a (summary value of SW for various values of k) and Figure 11b (average value of the CL/SG ratio for the considered values of k).

The analysis of the obtained results clearly indicates great benefits associated with the multi-level implementation of

TABLE 1. Experimental results – includes SW values.

	Benchmark			k=3			k=5			k=7		
	Name	In	Out	Cl	SG	Cl/SG	Cl	SG	Cl/SG	Cl	SG	Cl/SG
1	bw_21_11_9	5	3	8,322	3,838	2,168	6,875	2,846	2,416	4,177	2,846	1,468
2	bw_27_26_25	5	3	4,001	1,994	2,007	3,714	1,894	1,961	2,915	1,894	1,539
3	bw_321	5	3	8,814	2,792	3,157	4,817	1,801	2,675	4,817	1,801	2,675
4	bw_432	5	3	7,839	2,893	2,710	4,026	1,902	2,117	3,868	1,902	2,034
5	ldd_mno	9	3	6,91	0,162	42,654	5,741	0,184	31,201	3,62	0,146	24,795
6	sao2_210	10	3	16,605	0,953	17,424	11,705	0,611	19,157	10,262	0,552	18,591
7	sao2_310	10	3	16,219	0,663	24,463	11,697	0,465	25,155	10,162	0,427	23,799
8	sao2_320	10	3	6,931	1,143	6,064	4,964	0,683	7,268	4,779	0,537	8,899
9	sao2_321	10	3	11,511	1,139	10,106	8,521	0,679	12,549	7,072	0,533	13,268
10	clip_432	10	3	28,247	5,424	5,208	19,585	3,132	6,253	16,443	2,431	6,764
11	rd53	5	3	10,995	9,623	1,143	7,205	7,059	1,021	6,306	4,974	1,268
12	rd73	7	3	30,373	28,7	1,058	20,729	16,261	1,275	14,993	12,746	1,176
13	sao2	10	3	17,089	3,115	5,486	12,296	2,246	5,475	10,758	2,07	5,197
14	sqn	7	3	15,023	4,828	3,112	9,829	3,058	3,214	8,86	2,511	3,528
15	Misex1_356	7	3	6,726	0,953	7,058	4,784	0,862	5,550	3,74	0,714	5,238
16	Misex1_367	7	3	6,302	0,962	6,551	4,423	0,871	5,078	3,491	0,871	4,008
17	Misex1_567	8	3	6,801	0,551	12,343	4,873	0,444	10,975	4,002	0,444	9,014
18	Misex1_3456	8	4	8,04	0,609	13,202	5,426	0,563	9,638	4,442	0,563	7,890
19	Misex1_3567	7	4	8,488	1,592	5,332	5,849	1,501	3,897	4,978	1,501	3,316
20	Misex1_4567	8	4	8,116	0,522	15,548	5,607215	0,475891	11,783	4,737	0,476	9,952
	SUM:			233,352	72,456		162,666	47,538		134,422	39,939	
	MEAN:					9,340			8,433			7,721

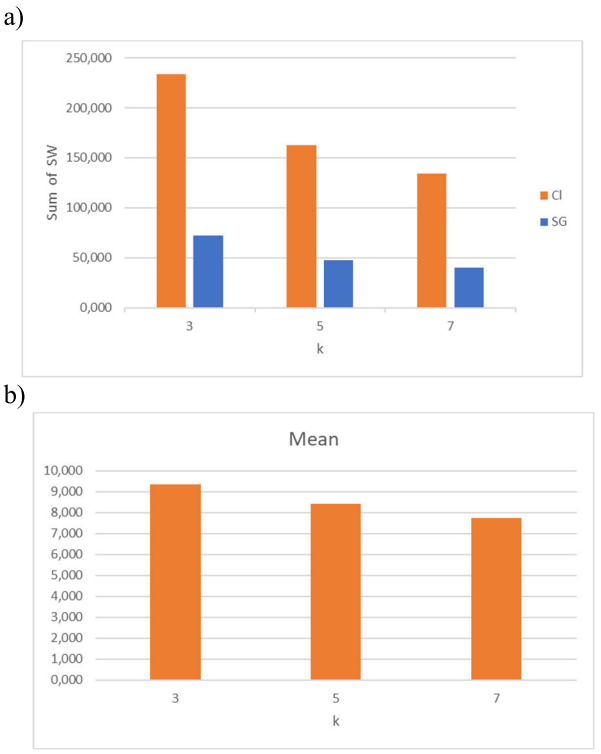

FIGURE 11. Graphs showing the experimental results: total SW value for both methods (a), average Cl/SG ratios (b).

logic functions described by graph of outputs. The total number of switches occurring in SOP networks is several times smaller compared to the classical implementation. The values of the total number of switches determined on the basis of switching graphs are much smaller than the number of switches in the classical solution for each value of k.

Greater benefits were obtained for smaller SOP blocks. For the considered set of benchmarks, in the case of the smallest SOP blocks (k = 3), the total number of switches decreased by over 9 times (CL/SG = 9.340), and for larger blocks (k = 7) - by over 7 times (CL/SG = 7.721). As the dynamic power is directly proportional to the SW value (eqn. 4), a doubling of the SW value results in a doubling of the dynamic power losses. Even though in the latest technologies the dynamic

power losses are lower than the static power losses, the obtained gain should be considered very significant.

The obtained dynamic power gain can also be expressed as a percentage. In the case of k = 3, the use of technology mapping based on the switching graph led to a reduction in switching activity by 68.9%. However, for k = 5 and k = 7, the following improvements in the results were obtained: 70.6% and 70.2%. It is also worth analyzing how often the use of a method based on a switching graph led to a significant improvement in the obtained results. It was assumed that a significant improvement is a situation in which Cl/SG > 2. Out of 20 test cases, for k = 3 in 18 cases the improvement was significant, for k = 5 in 17 cases the improvement was significant. However, for k = 7, significant improvement was achieved in 16 cases. It can also be noted that particularly good results were obtained for ldd_mno (Cl/SG > 42) and sao2. The worst results were obtained for the rd53 and rd73 functions.

VIII. CONCLUSION AND FUTURE WORK

Minimizing power consumption is the most important problem of modern logic synthesis. Energy savings are sought at every stage of synthesis. Although the greatest benefits should be sought at the level of the concept of describing the entire system, the search for energy-efficient concepts for the technology mapping of logic functions is also important. The paper shows the methodology of technology mapping leading to the reduction of the switching activity of combinational circuits. The proposed solution is an extension of the concept of mapping a multi-output function in the form of SOP networks described using an graph of outputs. An innovative idea is the concept of describing the mapping in the form of a switching graph. The proposed method of describing technology mapping in the form of a switchability graph makes it possible to easily combine combination modules into a larger whole and determine the total SW value for them.

The experimental results presented in the article clearly indicate the great potential of the proposed multi-output function mapping method in the form of SOP networks. Naturally, it is easy to imagine cases in which the proposed approach will not be beneficial. It should be emphasized, however, that the presented method can complement the classical method, which can work well in the case of functions for which minimization of ensembles does not lead to finding groups of multi-output implicants, which are naturally shared elements of several single-output functions, visible in the form of nodes in the graph of outputs.

Looking at the results obtained, it can be assumed that the use of a switching graph in the technology mapping process leads to an improvement in the results obtained in terms of switching activity by 70% compared to the classical method. This will lead to a significant reduction in the dynamic power of the circuit. The key element of this article was the assumption that the use of a switching model in the form of an innovative switching graph would lead to a reduction in the number of switches of the logic network. It can be seen that this assumption was confirmed. However, looking at it more critically, we can say that not all benchmarks have equally good results. It is difficult to indicate any clear rule on the basis of which one can determine the type of logic functions for which the presented ideas fit better or worse.‘

It is worth noting that the presented solutions may be subject to further optimization in terms of minimizing the total SW coefficient. This becomes possible by moving implicants between the levels of SOP networks and modifying the way connections are made between network elements. Generally speaking, the essence of optimization should consist in conducting SOP connections in such a way that the terms for which we obtain higher SW values are located in SOPs located closer to the outputs of the obtained system.

The planned directions of further work include the development of optimization techniques for SOP networks enabling the minimization of the total SW coefficient and the use of the developed methods in the FSM synthesis process. In the case of FSM, it will be possible to use a probabilistic description of automata, which will result in the possibility of predicting various probabilities of the value 1 appearing in the input nodes of the network. This fact will create significantly greater opportunities for optimizing SOP networks, due to greater possibilities of term shifts between SOP networks.

REFERENCES

[1] A. Raghunathan, N. K. Jha, and S. Dey, *High-Level Power Analysis and Optimization*. Boston, MA, USA: Springer, 1998.
[2] R. Bondade, "Hardware-software co-design of an embedded power management module with adaptive on-chip power processing schemes," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2010, pp. 617–620.
[3] L. Benini and G. De Micheli, "System-level power optimization: Techniques and tools," *ACM Trans. Design Autom. Electron. Syst.*, vol. 5, no. 2, pp. 115–192, 2000.
[4] P.-A. Hsiung and C.-W. Liu, "Exploiting hardware and software low power techniques for energy efficient co-scheduling in dynamically reconfigurable systems," in *Proc. Int. Conf. Field Program. Log. Appl.*, Aug. 2007, pp. 165–170.
[5] H. Ali and B. M. Al-Hashimi, "Architecture level power-performance tradeoffs for pipelined designs," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 1791–1794.
[6] O. S. Unsal and I. Koren, "System-level power-aware design techniques in real-time systems," *Proc. IEEE*, vol. 91, no. 7, pp. 1055–1069, Jul. 2003.
[7] S. Sinha and S. Tripathi, "BDD based logic synthesis and optimization for low power comparator circuit," in *Proc. Int. Conf. Intell. Circuits Syst.*, Phagwara, India, Apr. 2018, pp. 37–41.
[8] S. Roy, M. Choudhury, R. Puri, and D. Z. Pan, "Polynomial time algorithm for area and power efficient adder synthesis in high-performance designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 5, pp. 820–831, May 2016.
[9] N. Wang, W. Zhong, C. Hao, S. Chen, T. Yoshimura, and Y. Zhu, "Leakage-power-aware scheduling with dual-threshold voltage design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 10, pp. 3067–3079, Oct. 2016.
[10] S. Bard and N. I. Rafla, "Reducing power consumption in FPGAs by pipelining," in *Proc. 51st Midwest Symp. Circuits Syst.*, Aug. 2008, pp. 173–176.
[11] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. 27th Int. Symp. Comput. Archit.*, Jun. 2000, pp. 83–94.
[12] M. Kuc, W. Sułek, and D. Kania, "Low power QC-LDPC decoder based on token ring architecture," *Energies*, vol. 13, no. 23, p. 6310, Nov. 2020.
[13] M. Kuc, W. Sułek, and D. Kania, "FPGA-oriented LDPC decoder for cyber-physical system," *Mathematics*, vol. 8, no. 5, pp. 1–15, 2020.

[14] S. Kim and J. Kim, "Low-power data representation," *Electron. Lett.*, vol. 36, no. 11, p. 958, 2000.

[15] C. Chen, A. Srivastava, and M. Sarrafzadeh, "On gate level power optimization using dual-supply voltages," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 5, pp. 616–629, Oct. 2001.

[16] J. C. Chi, H. H. Lee, S. H. Tsai, and M. C. Chi, "Gate level multiple supply voltage assignment algorithm for power optimization under timing constraint," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 6, pp. 637–648, Jun. 2007, doi: 10.1109/tvlsi.2007.898650.

[17] A. Manzak and C. Chakrabarti, "A low power scheduling scheme with resources operating at multiple voltages," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 1, pp. 6–14, Feb. 2002.

[18] R. Mukherjee and S. O. Memik, "Evaluation of dual VDD fabrics for low power FPGAs," in *Proc. Conf. Asia South Pacific Design Autom. (ASP-DAC)*, 2005, p. 1240.

[19] Y. Shin, J. Seomun, K.-M. Choi, and T. Sakurai, "Power gating: Circuits, design methodologies, and best practice for standard-cell VLSI designs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 15, no. 4, pp. 1–37, Sep. 2010.

[20] N. Wang, W. Zhong, S. Chen, Z. Ma, X. Ling, and Y. Zhu, "Power-gating-aware scheduling with effective hardware resources optimization," *Integration*, vol. 61, pp. 167–177, Mar. 2018.

[21] M. Lin, H. Sun, and S. Kimura, "Power-efficient and slew-aware three dimensional gated clock tree synthesis," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, Tallinn, Estonia, Sep. 2016, pp. 1–6.

[22] M. R. Alam, M. E. S. Nasab, and S. M. Fakhraie, "Power efficient high-level synthesis by centralized and fine-grained clock gating," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 12, pp. 1954–1963, Dec. 2015, doi: 10.1109/TCAD.2015.2445734.

[23] E. Bezati, S. Casale-Brunet, M. Mattavelli, and J. W. Janneck, "Clock-gating of streaming applications for energy efficient implementations on FPGAs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 4, pp. 699–703, Apr. 2017, doi: 10.1109/TCAD.2016.2597215.

[24] R. Mullins and S. Moore, "Demystifying data-driven and Pausible clocking schemes," in *Proc. 13th IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, Mar. 2007, pp. 175–185.

[25] Y. S. Park, Y. Tao, and Z. Zhang, "A fully parallel nonbinary LDPC decoder with fine-grained dynamic clock gating," *IEEE J. Solid-State Circuits*, vol. 50, no. 2, pp. 464–475, Feb. 2015, doi: 10.1109/JSSC.2014.2362854.

[26] F. Xin, M. Krstić, and E. Grass, "Improvements of Pausible clocking scheme for high-throughput and high-reliability GALS systems design," in *Advanced Techniques in Logic Synthesis, Optimizations and Applications*. New York, NY, USA: Springer, 2011, pp. 401–417.

[27] R. S. Shelar, H. Narayanan, and M. P. Desai, "Orthogonal partitioning and gated clock architecture for low power realization of FSMs," in *Proc. 13th Annu. IEEE Int. ASIC/SOC Conf.*, Sep. 2000, pp. 266–270.

[28] W. Shen, Y. Cai, X. Hong, and J. Hu, "Activity-aware registers placement for low power gated clock tree construction," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Sep. 2007, pp. 383–388.

[29] K. Saleh, M. Najibi, M. Naderi, H. Pedram, and M. Sedighi, "A novel clock generation scheme for globally asynchronous locally synchronous systems: An FPGA-validated approach," in *Proc. 15th ACM Great Lakes Symp. (VLSI)*, Apr. 2005, pp. 296–301.

[30] E. Amini, M. Najibi, and H. Pedram, "Globally asynchronous locally synchronous wrapper circuit based on clock gating," in *Proc. IEEE Comput. Soc. Annu. Symp. Emerg. VLSI Technol. Archit. (ISVLSI)*, Mar. 2006, pp. 193–199.

[31] M. Krstic, E. Grass, F. K. Gürkaynak, and P. Vivet, "Globally asynchronous, locally synchronous circuits: Overview and outlook," *IEEE Des. Test Comput.*, vol. 24, no. 5, pp. 430–441, Sep. 2007.

[32] A. Kulmala, T. Hamalainen, and M. Hannikainen, "Reliable GALS implementation of MPEG-4 encoder with mixed clock FIFO on standard FPGA," in *Proc. Int. Conf. Field Program. Log. Appl.*, 2006, pp. 1–6.

[33] Z. Yu and B. M. Baas, "High performance, energy efficiency, and scalability with GALS chip multiprocessors," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 66–79, Jan. 2009, doi: 10.1109/tvlsi.2008.2001947.

[34] D. Modrzyk and D. Kania, "Asynchroniczna wymiana danych W układzie GALS ukierunkowana na minimalizacje poboru mocy," *Przegląd Elektrotechniczny*, vol. 90, no. 2, pp. 132–137, 2014.

[35] T. Mendez and S. G. Nayak, "Design of a low-power computational unit using a pipelined Vedic multiplier," in *Proc. Int. Conf. Advancement Technol. (ICONAT)*, Goa, India, Jan. 2023, pp. 1–6.

[36] Y. Aghaghiri, F. Fallah, and M. Pedram, "Transition reduction in memory buses using sector-based encoding techniques," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 8, pp. 1164–1174, Aug. 2004.

[37] L. Benini, G. De Micheli, A. Lioy, E. Macii, G. Odasso, and M. Poncino, "Synthesis of power-managed sequential components based on computational kernel extraction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 9, pp. 1118–1131, Sep. 2001.

[38] A. Barkalov, L. Titarenko, and S. Chmielewski, "Improving characteristics of LUT-based Moore FSMs," *IEEE Access*, vol. 8, pp. 155306–155318, 2020.

[39] A. Barkalov, L. Titarenko, and K. Mielcarek, "Improving characteristic of LUT-based Mealy FSMs," *Int. J. Appl. Math. Comput. Sci.*, vol. 30, no. 4, pp. 745–759, 2020.

[40] Y.-H. Chen, T.-C. Chen, C.-Y. Tsai, S.-F. Tsai, and L.-G. Chen, "Algorithm and architecture design of power-oriented H.264/AVC baseline profile encoder for portable devices," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp. 1118–1128, Aug. 2009.

[41] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "Power reduction techniques for LDPC decoders," *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1835–1845, Aug. 2008, doi: 10.1109/JSSC.2008.925402.

[42] Y. S. Park, D. Blaauw, D. Sylvester, and Z. Zhang, "Low-power high-throughput LDPC decoder using non-refresh embedded DRAM," *IEEE J. Solid-State Circuits*, vol. 49, no. 3, pp. 783–794, Mar. 2014, doi: 10.1109/JSSC.2014.2300417.

[43] G. Yong, Y. Yuan, and F. Bei, "The application of bus coding in designing of low power SoC based on wishbone," in *Proc. Int. Conf. Comput. Sci. Inf. Process. (CSIP)*, Aug. 2012, pp. 1132–1135, doi: 10.1109/CSIP.2012.6309057.

[44] K. Khatua, S. Chattopadhyay, and B. Sahoo, "Random forest regression model AugmenteD with KL algorithm (RFRKL) to minimize power in FSM synthesis," in *Proc. Int. Interdiscipl. Conf. Math., Eng. Sci. (MESICON)*, Durgapur, India, Nov. 2022, pp. 1–6.

[45] S. Chattopadhyay, P. Yadav, and R. K. Singh, "Multiplexer targeted finite state machine encoding for area and power minimization," in *Proc. IEEE INDICON 1st India Annu. Conf.*, Jun. 2004, pp. 12–16.

[46] K. Kajstura and D. Kania, "Dekompozycyjna metoda kodowania stanów wewnętrznych automatu skończonego ukierunkowana na minimalizację mocy," *Przegląd Elektrotechniczny*, vol. 87, no. 6, pp. 146–150, 2011.

[47] K. Kajstura and D. Kania, "Binary tree-based low power state assignment algorithm," in *Proc. 12th Int. Conf. Comput. Methods Sci. Eng.*, Athens, Greece, Mar. 2016, pp. 17–20.

[48] K. Kajstura and D. Kania, "Low power synthesis of finite state machines—State assignment decomposition algorithm," *J. Circuits, Syst. Comput.*, vol. 27, no. 3, Mar. 2018, Art. no. 1850041.

[49] L. Mengibar, L. Entrena, M. G. Lorenz, and E. S. Millan, "Patitioned state encoding for low Power in FPGAs," *Electron. Lett.*, vol. 41, pp. 948–949, Aug. 2005.

[50] Y. Xia and A. E. A. Almaini, "Genetic algorithm based state assignment for power and area optimisation," *IEE Proc. Comput. Digit. Techn.*, vol. 149, no. 4, p. 128, 2002.

[51] K. Khatua, "Levenberg–Marquardt backpropagation model augmented with Prim's algorithm approach (LMBP) to minimize power in FSM synthesis," in *Proc. IEEE Int. Conf. Semiconductor Electron. (ICSE)*, Kuala Lumpur, Malaysia, Aug. 2022, pp. 117–120.

[52] C. Lokesh, M. R. Supritha, M. Kumar, K. Rekha, and P. Das, "Dynamic power synthesis techniques for combinational circuit," in *Proc. Int. Conf. Recent Adv. Sci. Eng. Technol. (ICRASET)*, Nagara, India, Nov. 2023, pp. 1–7.

[53] M. Kubica, A. Opara, and D. Kania, "Logic synthesis strategy oriented to low power optimization," *Appl. Sci.*, vol. 11, no. 19, p. 8797, Sep. 2021.

[54] A. Opara, M. Kubica, and D. Kania, "Decomposition approaches for power reduction," *IEEE Access*, vol. 11, pp. 29417–29429, 2023.

[55] A. Opara and M. Kubica, "Technology mapping of multi-output functions leading to the reduction of dynamic power consumption in FPGAs," *Int. J. Appl. Math. Comput. Sci.*, vol. 33, no. 2, pp. 267–284, 2023.

[56] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual for System-on-Chip Design*. Springer, 2007.

[57] Micheli G., *Synthesis and Optimization of Digital Circuits*. New York, NY, USA: McGraw-Hill, 1994.

[58] D. Kania, "Logic decomposition for PAL-based CPLDs," *J. Circuits, Syst. Comput.*, vol. 24, no. 3, Mar. 2015, Art. no. 1550042.

[59] D. Kania, "An efficient approach to synthesis of multi-output Boolean functions on PAL-based devices," *IEEE Proc. Comput. Digit. Techn.*, vol. 150, no. 3, pp. 143–149, May 2003.
[60] D. Kania, "A technology mapping algorithm for PAL-based devices using multi-output function graphs," in *Proc. 26th Euromicro Conf. EUROMICRO Inform., Inventing Future*, vol. 1, Jun. 2000, pp. 146–153.
[61] D. Kania, "A new approach to logic synthesis of multi-output Boolean functions on pal-based CPLDS," in *Proc. 17th ACM Great Lakes Symp. (VLSI)*, Mar. 2007, pp. 152–155.
[62] M. Kubica, D. Kania, and J. Kulisz, "A technology mapping of FSMs based on a graph of excitations and outputs," *IEEE Access*, vol. 7, pp. 16123–16131, 2019.
[63] M. Kubica and D. Kania, "Graph of outputs in the process of synthesis directed at CPLDs," *Mathematics*, vol. 7, no. 12, p. 1171, Dec. 2019.
[64] (2008). *Collaborative Benchmarking and Experimental Algorythmics Laboratory*. [Online]. Available: https://www.cbl.ncsu.edu:16080/benchmarks/LGSynth93/testcase/

MARCIN KUBICA received the M.Sc. and Ph.D. degrees from the Silesian University of Technology, Gliwice, Poland, in 2010 and 2014, respectively. He has been an Assistant Professor with the Institute of Electronics, Silesian University of Technology, and has been a Professor, since 2022. His main research interests include programmable devices and systems and logic synthesis.

BOLESŁAW POCHOPIEŃ received the Ph.D. and D.Sc. degrees from the Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Gliwice, Poland. He is a Professor with the Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology. His scientific interests include informatics and automatic control, in practical theory and implementation techniques of logic circuits and arithmetic of logic systems.

DARIUSZ KANIA received the Ph.D. degree from the Silesian University of Technology, Gliwice, Poland. He has been a Professor, since 2007. His research interests include programmable devices and systems, logic synthesis, technology-dependent partitioning, technology mapping dedicated to a wide range of programmable logic devices (FPGA and pSoC), cyber-physical systems, globally asynchronous locally synchronous digital circuits, hardware description languages (Verilog and VHDL), low-power implementation of digital circuits, posturography in postural control diagnostics, motor functions rehabilitation, and music information retrieval. He is a member of the Electronics and Telecommunication Committee, Polish Academy of Sciences.