electronics

MDPI

Article

Multi-Level Sum of Product (SOP) Network Power Optimization Based on Switching Graph

Marcin Kubica * and Dariusz Kania

Department of Digital Systems, Silesian University of Technology, ul. Akademicka 2A, 44-100 Gliwice, Poland; dkania@polsl.pl

* Correspondence: mkubica@polsl.pl

Abstract: The article presents the methodology of optimization of technology mapping of a multi-output function implemented in the form of sum of product (SOP) networks. The optimization is based on the concept of reducing the switching activity of combinational circuits. The aim of reducing network switching is to limit the consumption of dynamic power. Since dynamic power is one of the components of the total power consumed by digital systems, this leads to an optimization focused on the energy efficiency of digital systems. The basis of the proposed method is the technology mapping using a modified output graph describing the result of minimizing the multi-output function. The modified output graph, in terms of parameters associated with dynamic power, is defined as a switching graph. It was assumed that the key parameter associated with dynamic power is the switching activity of individual nodes of the logic network. The article presents elements of switching graph optimization which lead to the improvement of parameters associated with the dynamic power of the circuit. The essence of the proposed optimization methods is the appropriate movement of products and connections occurring in the logic network. Reducing the number of logic network vertices is also extremely important. The effectiveness of the proposed method was confirmed by the results of experiments performed on selected benchmarks. A significant reduction in the total value of switching activity was obtained for the optimized structures.

Keywords: low-power synthesis; SOP; switching activity; technology mapping

check for updates

Citation: Kubica, M.; Kania, D. Multi-Level Sum of Product (SOP) Network Power Optimization Based on Switching Graph. *Electronics* **2024**, *13*, 4011. https://doi.org/10.3390/electronics13204011

Academic Editor: Stefano Ricci

Received: 21 August 2024
Revised: 3 October 2024
Accepted: 9 October 2024
Published: 12 October 2024

1. Introduction

The issue of designing energy-efficient electronic devices [1] has become an extremely important issue. Especially in the area of digital systems, it is particularly important to optimize power consumption in selected stages of logic synthesis. The basis of this approach is to maximize the reliability of such systems and to ensure the greatest possible mobility of digital devices. Both of these attributes are extremely important in cyber-physical systems (CPSs), where reliable autonomous operation must be combined with the maximization of battery operation [2]. Greater reliability is achieved by reducing the complexity of the logic structure and limiting the amount of heat that must be dissipated. It also means avoiding thermal hot spots as potential points of damage. The second attribute—mobility—is de facto a time parameter. It determines the time of effective operation of a given device under battery power conditions. Cyber-physical systems are widely used in medicine, military, and space applications and require a number of mechanisms that allow for an effective reduction in power consumption.

Mechanisms leading to a reduction in power consumption by a digital system can be divided into two groups, those implemented at high-level synthesis stages and those implemented at low-level synthesis stages.

In the first case, a natural method of reducing power consumption is to divide tasks between hardware and software implementation, or temporarily disable selected modules [3–5]. It turns out that limiting power consumption in high-level synthesis processes can be extremely effective [3,5–10]. A number of articles address this problem [7,11–15]. Similarly, in the case of low-level synthesis, there are methods leading to reducing power consumption. The most important ideas leading to reducing power consumption include the idea of locally reducing the supply voltage [16–19], "power gating" [13–24], or temporarily blocking the clock signal (clock gating) [25–31]. Alternatively, to block the clock signal, it is sometimes better to only lower the clock frequency in some parts of the circuit [32,33]. Similarly, there is a strategy based on an asynchronous implementation of the circuit, with only small parts of the circuit operating in a synchronous way (globally asynchronous locally synchronous) [34–39]. In addition to concepts targeted at specific circuit applications [40–43], there are more general approaches in low-level synthesis. In the case of sequential circuits, such an idea may be an efficient implementation of an FSM (finite state machine) [44–46] or the appropriate encoding of internal states [47–50]. In the case of combinational circuits, a good example of such an idea is the appropriately conducted decomposition of a multi-output function [51–53], which leads to a reduction in the number of switches of the implemented circuit.

The above literature review clearly indicates that reducing power consumption is an extremely important issue. This also applies in the case of CPSs, where the developed strategies for their implementation [54–57] can be expanded with a component associated with energy saving.

The aim of the article is to present the authors' methods of optimizing logic structures leading to the reduction in the circuit's switching activity, and consequently to the reduction in power consumption. This article is a continuation of a series of articles in which the authors discuss the topic of reducing power consumption in digital systems, which is why the authors often refer to earlier articles to explain some concepts [48,51–53,58]. The paper [48] presents the ideas of minimizing power consumption in sequential circuits by the appropriate encoding of internal states. The papers [51–53] describe the ideas of reducing switching activity associated with FPGA-oriented decomposition. The paper [58] is particularly important and presents the general concept of realizing multi-output functions in the form of energy-efficient SOP networks. The switching graph concepts are introduced there, and the obtained technology mapping results are compared with the classical method.

This paper extends the approach from [58] with an optimization element such as moving terms between SOP networks nodes, moving connections in SOP networks, and splitting (removing) network vertices. The obtained mapping results are compared with the approach without optimization from [58], as well as with the classical approach.

The article is divided into sections. Section 2 presents the basic concepts necessary to discuss the ideas presented in this article. Section 3 presents an example, based on which the ideas leading to the reduction in the switching activity of the SOP circuit are discussed, such as shifting terms between SOP networks nodes or shifting connections in SOP networks. Section 4 proposes an algorithm for technology mapping that takes into account new ideas. Section 5 contains the experimental results confirming the effectiveness of the proposed methods. The article ends with a summary.

2. Theoretical Background

Minimizing power consumption is an extremely important problem in the synthesis of modern digital systems. The power consumed by a digital device contains two components, namely static power (P_{stat}) and dynamic power (P_d). Static power is closely related to the technology in which the considered system was made, and the designer has no influence on its size. Dynamic power is related to the conditions in which the considered system operates, and the user has a significant influence on its value.

In new technologies, the potential possibilities of minimizing power losses lean towards static power, but it is still important to search for synthesis methods that allow for minimizing dynamic power. This is because all potential possibilities of minimizing power losses are used, and minimizing dynamic power can be achieved with relatively simple procedures. The essence of such minimization is a properly conducted logic synthesis process or optimization in terms of power of the technology mapping process.

The basis of dynamic power losses (P_d) is the circuit switching process and the related need to overload internal capacities. Dynamic power can be described by Relationship (1) as follows:

$$P_d = \sum_{i=1}^{n} \frac{1}{2T} SW_i \cdot C_i \cdot V_{dd}^2 \tag{1}$$

where n is the number of nodes of the analyzed logic network, T is the time for which we determine power losses (the signal at the outputs of individual gates is usually not periodic; therefore, depending on the dynamic power, the time T is used instead of the fclk frequency), SW_i is the switching activity of the ith node, C_i is the capacitance associated with the ith node, and V_{dd} is the supply voltage.

The dynamic power consumed by the system is therefore proportional to the square of the supply voltage, load capacitance, and switching activity. The value of V_{dd} and C_i depends on the technology of the system, so the only element that can be minimized in the synthesis process is the switching activity. When minimizing dynamic power losses, it is necessary to ensure that the logic states in individual nodes of the system change as rarely as possible. The value of SW_i is directly related to the probability of the occurrence of value 1 in a given node of the system. Knowing the probability of the occurrence of value 1 in the *ith* node of the system, which we assume is equal to $p(x)$, it becomes possible to determine the switching activity in this node according to Relationship (2).

$$SW_i = 2p(x)(1 - p(x)) \tag{2}$$

Knowing the values of the probabilities $p(x)$ and the occurrence of the value 1 for individual inputs of gates or larger logic blocks, it becomes possible to determine the probabilities $p(y)$ at their outputs.

Let us consider the frequently encountered SOP blocks, which are a circuit representation of a function in the form of a sum of products. Let us assume that these blocks contain k terms with n inputs. The value of the probability $p(y)$ at the output of the SOP block in such a case is expressed by Relationship (3).

$$p(y) = 1 - \prod_{j=1}^{k} (1 - p_{AND}(j)) \tag{3}$$

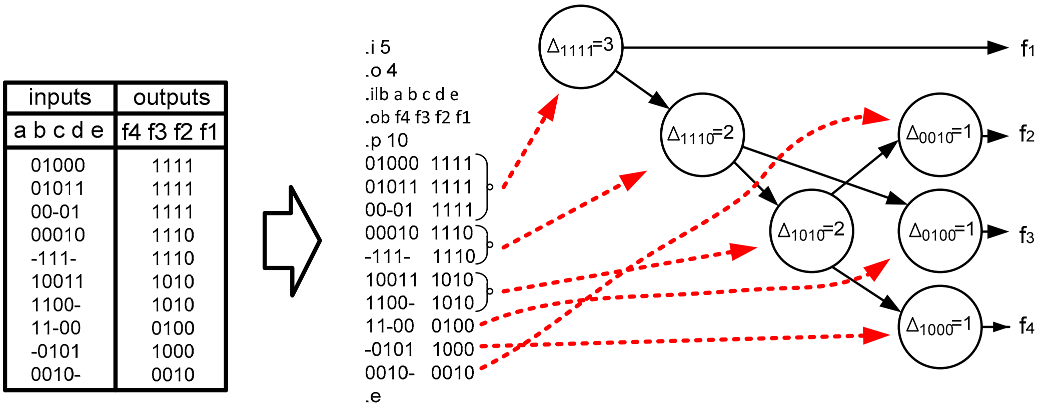
where $p_{AND}(j) = \prod_{i=1}^{n} (p(i))$.

The technology representation of the minimized form of the function f: $B^n \to B^m$, where $B = \{0, 1\}$ can be symbolically represented in the form of an output graph [59,60]. The vertices of this graph, which correspond to the output parts of multi-output implicants, can be associated with the discriminants Δ_{y_i}, where y_i determines the m element output part of the ith multi-output implicant. The output graph $G < Y, \overrightarrow{U} >$ is a directed graph, where Y is the set of all vertices Δ_{y_i}, while $\overrightarrow{U}$ the set of edges connecting the vertices $\Delta_{y_{si}}$, $\Delta_{y_{ri}}$ such that the code distance of the vectors y_{si}, y_{ri} is 1 and $\mu(\Delta_{y_{si}}) + 1 = \mu(\Delta_{y_{ri}})$, where μ is called the order of the vertex and determines the number 1 in the output part of the implicant.

The result of minimizing the example multi-output function and the corresponding output graph are presented in Figure 1.

Each vertex of the output graph can be directly mapped into the SOP structure. As a result of such mapping, a network of SOP blocks is created, shown symbolically in Figure 2.

Each SOP network node is associated with an appropriate set of implicants mapped as products of an appropriate number of literals/variables. The numbers of variables are represented in Figure 2 as numbers placed next to arrows pointing towards SOP network nodes.



inputs	outputs
a b c d e	f4 f3 f2 f1
01000	1111
01011	1111
00-01	1111
00010	1110
-111-	1110
10011	1010
1100-	1010
11-00	0100
-0101	1000
0010-	0010

```
.i 5
.o 4
.ilb a b c d e
.ob f4 f3 f2 f1
.p 10
01000  1111
01011  1111
00-01  1111
00010  1110
-111-  1110
10011  1010
1100-  1010
11-00  0100
-0101  1000
0010-  0010
.e
```

Figure 1. The result of minimizing the example function *f*: $B^5 \rightarrow B^4$ and its corresponding output graph.

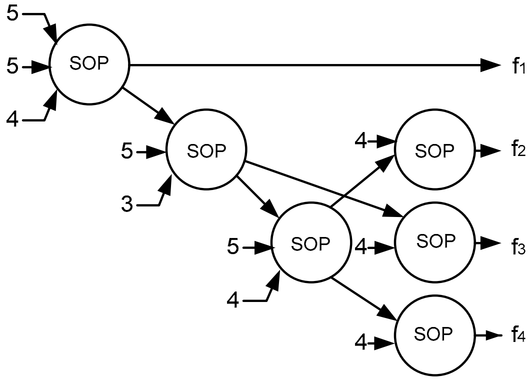


Figure 2. The structure of the SOP network representing the output graph of the example function *f*: $B^5 \rightarrow B^4$.

For each SOP, the switching activity coefficient and the probability of the value 1 appearing at the SOP output can be determined according to Relationship (2). Knowing the expressions implemented by all SOP blocks, primarily the number of literals for individual terms, it becomes possible to determine the switching activity of the entire SOP network.

Assuming that the input variables (a, b, c, d, e) are independent and that the value 1 at individual inputs is as probable as the value 0, it is possible to determine the probability of the value 1 appearing at the outputs of AND gates and the output of the OR gate. This information entered into the output graph leads to the creation of the switching graph described in [58]. For the considered case, the switching graph shown in Figure 3 was obtained.

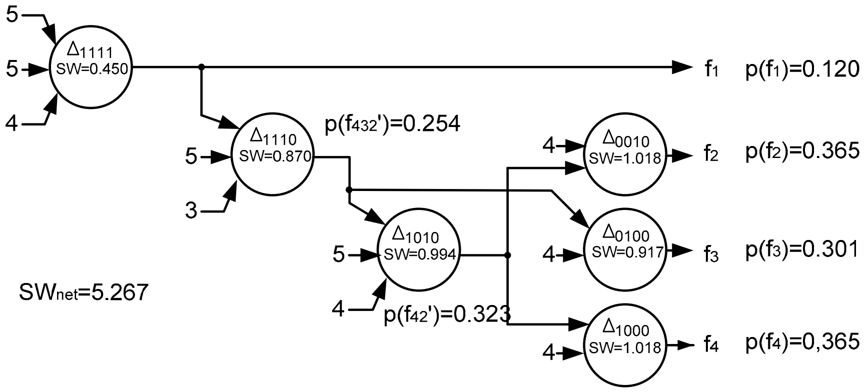


Figure 3. Switching graph for the example function *f*: $B^5 \rightarrow B^4$.

Assuming that the SOP blocks contain four terms, the mapping of the switching graph from Figure 3 leads to the logic structure shown in Figure 4.

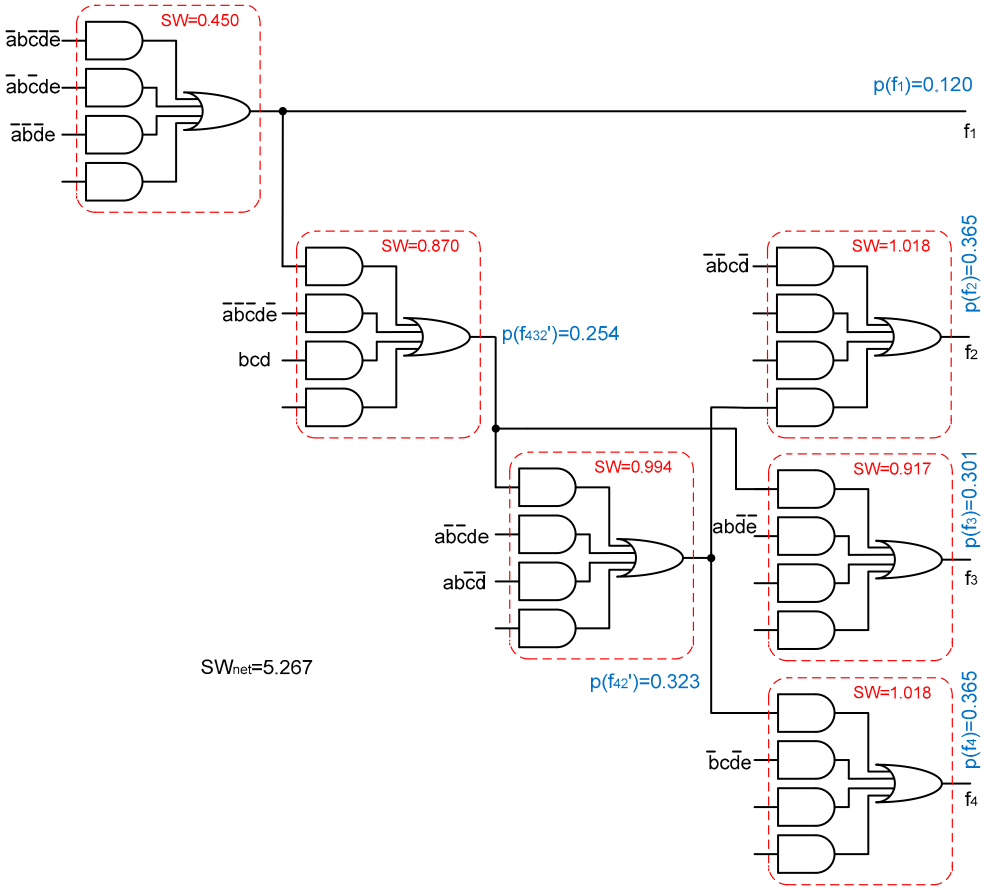


Figure 4. Technology mapping of the considered multi-output function f: $B^5 \to B^4$ obtained using a switching graph.

3. SOP Network Optimization Strategy for SW Minimization

It turns out that it is possible to optimize the technology mapping of the considered multi-output function, the multi-level system implementation of which is presented in Figure 4 in terms of the number of switches. The essence of optimization should consist of locating the products for which the highest values of the SW coefficient occur as close as possible to the outputs. Applying such a strategy reduces the number of switches in the entire network, because the less frequent switching of blocks located closer to the inputs affects the minimization of switching of SOP network nodes located at levels associated with higher-order vertices. In this situation, the algorithm of action should consist in shifting terms or connections between nodes, i.e., network levels, towards levels closer to the outputs. Of course, shifting cannot affect the logic functions implemented by the optimized SOP networks.

In the following part of this section, the elements of the optimization strategy will be presented using the example considered in the previous section. The example was chosen in such a way as to enable the visualization of all the elements available in the optimization process. The essence of optimization lies in the sequential application of one of the three optimization methods, which are as follows:

(i) Moving terms between SOP networks nodes;
(ii) Moving connections in SOP networks;
(iii) Splitting (removing) network vertices, which can lead to a reduction in nodes.

All node shifts and splits are inextricably linked to the use of unused products contained in SOP network nodes located closer to the outputs. In each optimization step, one of the above optimization possibilities is selected. The selection is determined by potentially greater benefits, which comes down to a greater minimization of the total SW value of the entire network. In each optimization step, the results will be presented in the form of a switching graph of the obtained network, for which the total value of the SOP network's (SW_{net}) switching coefficient was determined.

The value of the SW_{net} coefficient for the initial form of SOP networks subjected to optimization (Figure 3), which is the sum of the SW coefficients for all SOP blocks, is $SW_{net\mathrm{t}}$ = 5.267.

Since the determinants of the lowest-level vertices of the switching graph shown in Figure 3, Δ_{1000} = Δ_{0100} = Δ_{0010} = 1, there are still unused products in the nodes located closest to the outputs. We assume that there are four terms (k = 4) in the SOP blocks. Therefore, in each node whose outputs are associated with the functions f2, f3, and f4, we have free products at our disposal.

The network analysis begins with the node associated with the vertex of the switching graph for which the discriminant Δ_{1111} = 3. Since the SOP output is associated with the output node of the f1 function, any shifts of the terms are not justified. The next node subject to analysis is the node associated with the discriminant Δ_{1110}. The node receives three signals that are associated with the functions implemented on the individual terms. The first one is the output from the previous node Δ_{1111}. According to Equation (3), the probability of the appearance of value 1 is $p(f1) = 1 - (1 - (0.5)^5)\ (1 - (0.5)^5)\ (1 - (0.5)^4) \cong 0.120$. The remaining input signals are associated with the products of five or three variables. Assuming the values of the probability of the appearance of value 1 for each variable are equal to 0.5, the probabilities of the appearance of value 1 at the output of the products are, respectively,

$$p\left(\bar{a}\bar{b}\bar{c}\mathrm{d}\bar{e}\right) = 0.5^5 = 0.03125$$

$$p(\mathrm{bcd}) = 0.5^3 = 0.125$$

According to the adopted principle (i), the term for which the SW coefficient takes the maximum value should be advantageously moved to a level closer to the network output signals. This condition is met by the term realizing the sum of three products, because according to the general dependence (2), the values of the coefficients of the individual terms satisfy the inequality

$$SW(\mathrm{bcd}) = 0.2188 > SW(p(f1)) = 0.2115 > SW\left(\bar{a}\bar{b}\bar{c}\mathrm{d}\bar{e}\right) = 0.0605$$

Since the product bcd is associated with the vertex of the switching graph associated with the discriminant Δ_{1110}, it is an implicant of the functions f4, f3, and f2. It is therefore possible to shift this product to the nodes associated with the vertices Δ_{1000}, Δ_{0100}, and Δ_{0010}. The shift result, which is the effect of the first step of the optimization consisting of shifting the terms between the network nodes, is illustrated in the switching graph of SOP networks presented in Figure 5.

The effect of the optimization is the reduction in the total value of the SW coefficient to the value of SW_{net} = 5.059. It is worth paying attention to the values of the SW coefficients associated with the individual SOP network nodes. The effect of the shifts is the increase in the SW coefficient values associated with the nodes closest to the outputs. More frequent switching of blocks located close to the outputs is more than compensated for by the reduction in the SW coefficient value associated with the node of the SOP network switching graph corresponding to the Δ_{1110} discriminant.

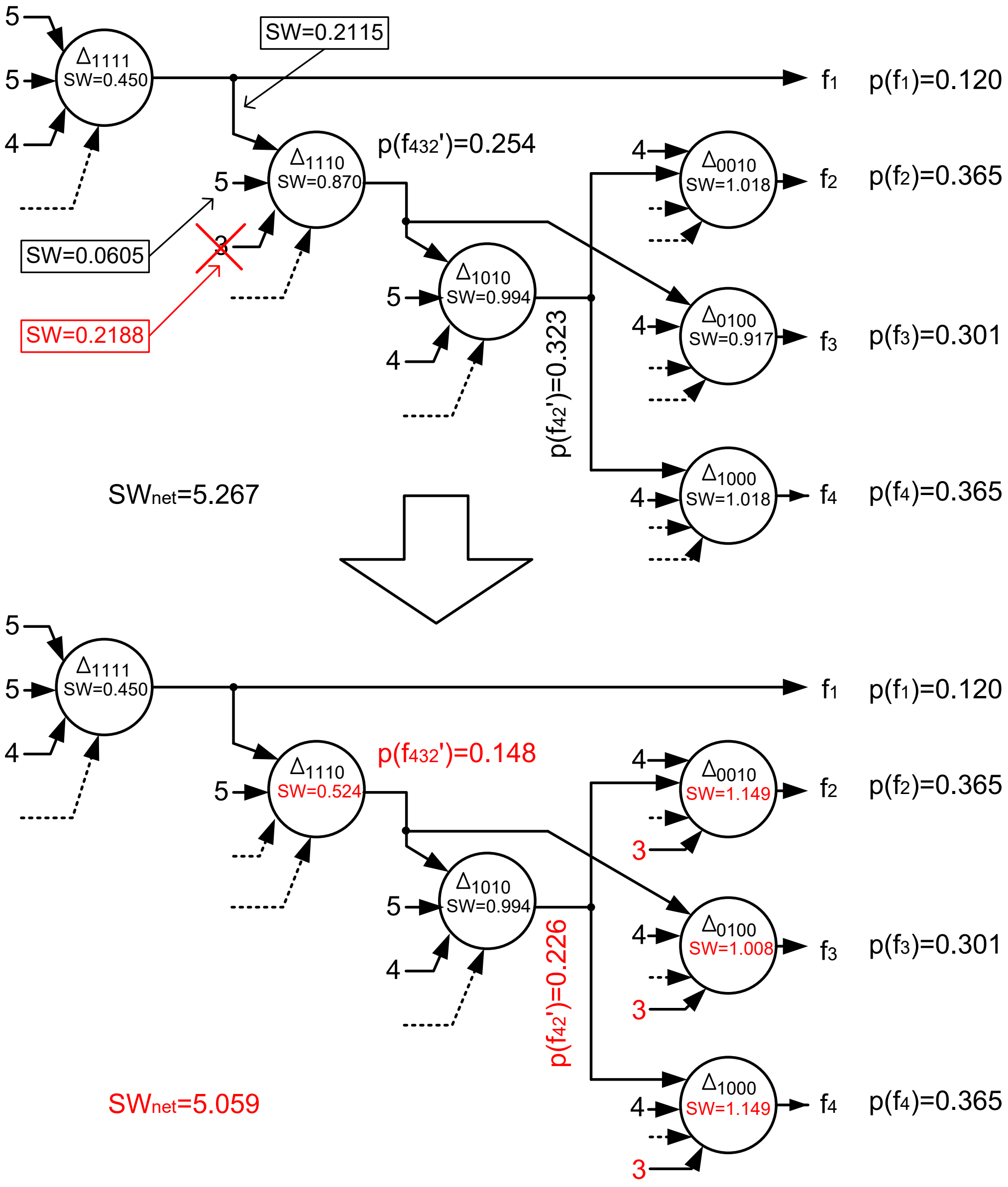


Figure 5. The first step of the SOP block network optimization—switching graphs of the network before and after optimization.

In the resulting graph (Figure 5), there are still unused products in the nodes associated with the discriminants Δ_{1000}, Δ_{0100}, and Δ_{0010}. Therefore, there is a possibility of further optimization, which comes down to shifting the products or connections from the levels closer to the inputs to the nodes associated with the outputs. Let us again focus on the node Δ_{1110}. Two signals reach it, of which the connection between the nodes with the SOP block associated with the vertex Δ_{1111} has a higher value of the SW coefficient. Therefore, it becomes possible to apply the second optimization principle (ii), which consists of shifting this connection directly to the nodes associated with the vertices Δ_{1000}, Δ_{0100}, and Δ_{0010}, which is illustrated in Figure 6.

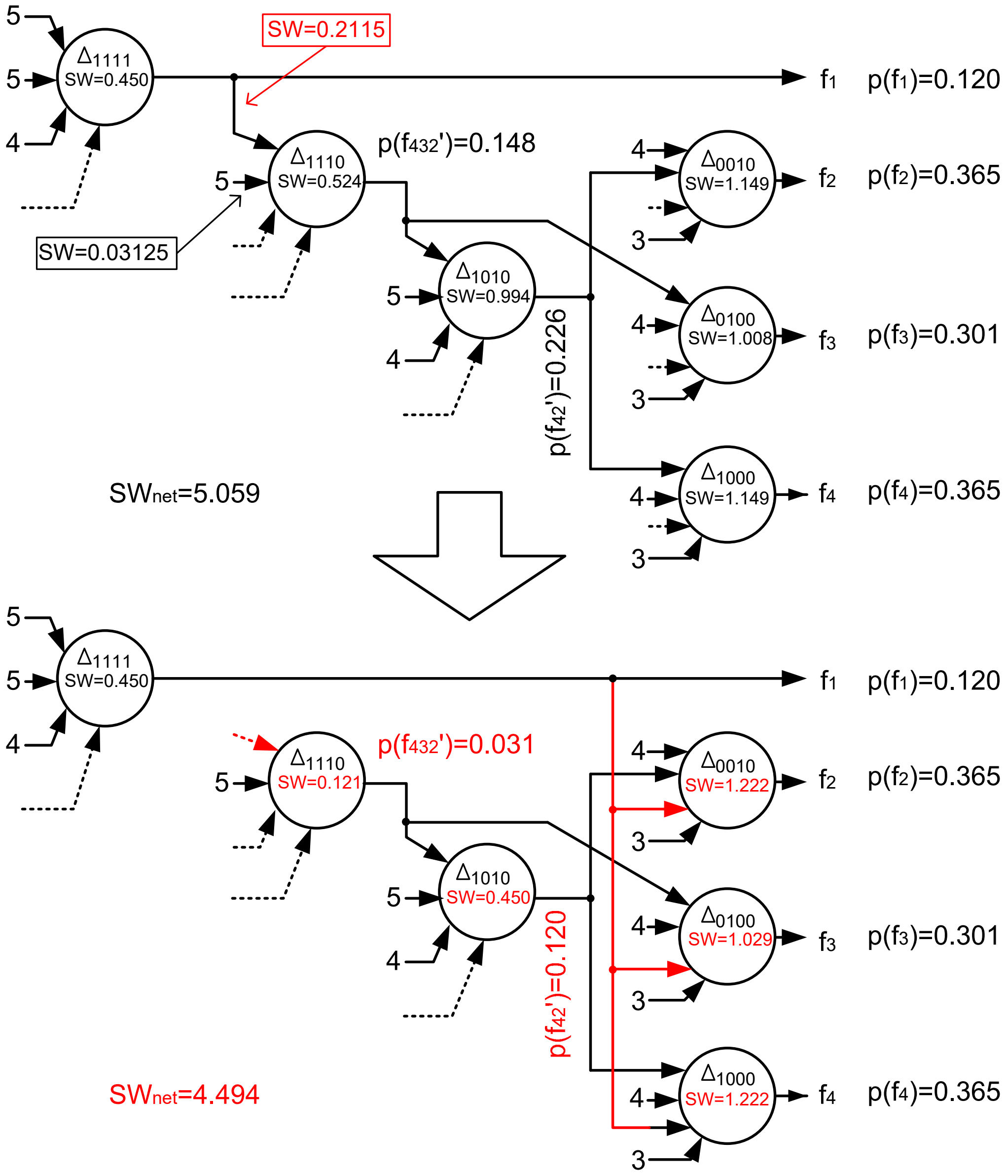


Figure 6. The second step of the SOP block network optimization—switching graphs of the network before and after optimization.

This led to a further reduction in the SW coefficient to the value of SW_{net} = 4.494. It is worth noting that the connection shifting process led to the use of all the SOP block terms that generate the output signals of the entire network. In the next optimization steps, attention should be focused on using the unused product of the SOP block associated with the vertex Δ_{1010}. This becomes possible by using the third optimization rule (iii), resulting from the possibility of splitting the vertex, which results in its reduction, as illustrated in Figure 7.

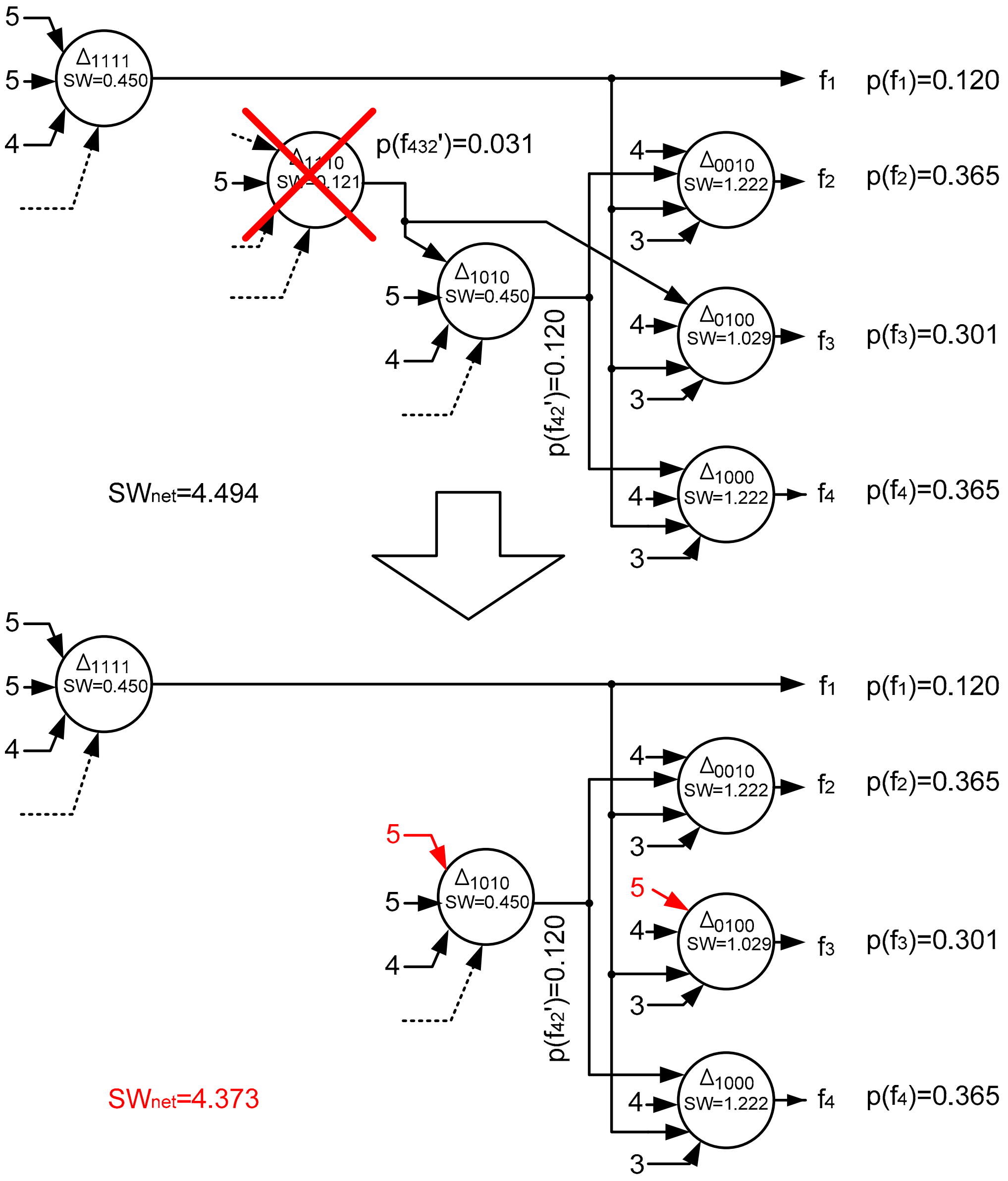


Figure 7. The third step of the SOP network optimization—network switching graphs before and after optimization.

It is worth noting that the obtained reduction in the Δ_{1110} vertex did not affect the SW coefficients of other SOP network nodes. The reduction in the SOP network coefficient value to the value of *SWnet* = 4.373 results only from the splitting of the the Δ_{1110} vertex, which led to the elimination of one SOP network node.

The network optimization process led to a 17% reduction in dynamic power, which is the effect of reducing the coefficient characterizing the SOP networks switching from the level of *SWnet* = 5.267 to the value of *SWnet* = 4.373. The final implementation of the system after optimization is shown in Figure 8.

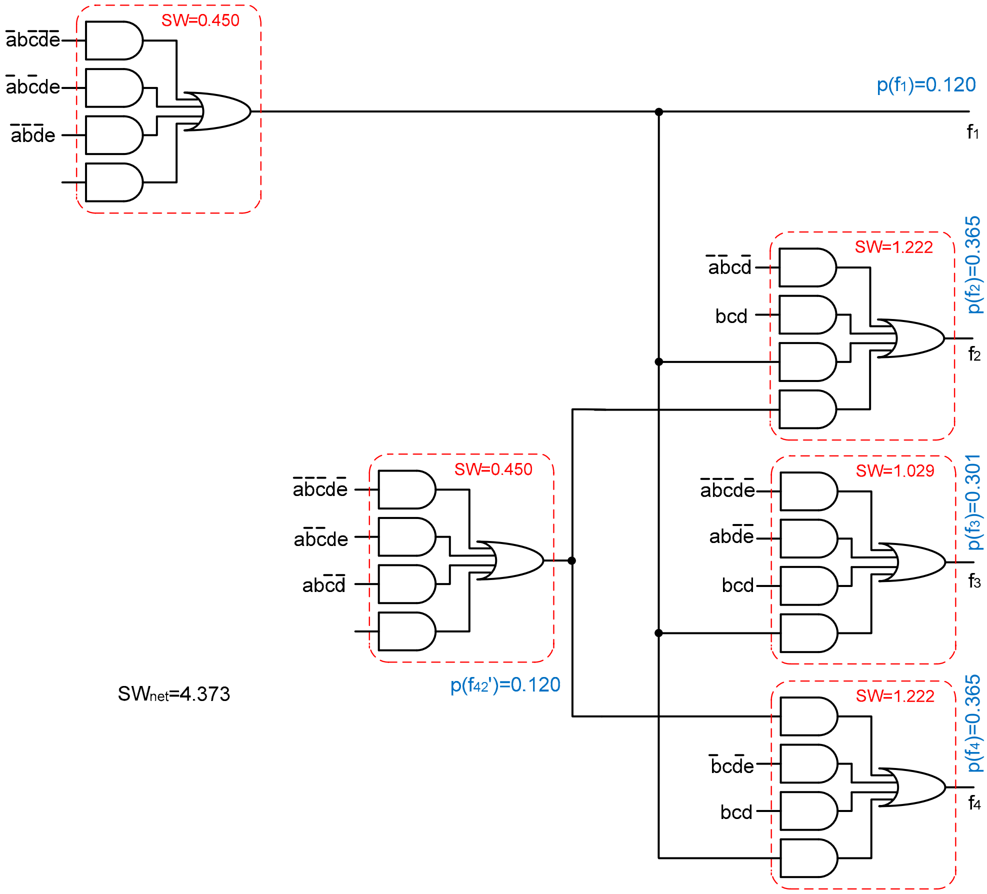


Figure 8. System implementation of the considered set of functions f: $B^5 \to B^4$ after optimization.

4. Algorithm for Technology Mapping Multi-Output Function Aimed at Reducing Switching SOP Networks

The methods of optimizing the mapping of a multi-output function in the form of SOP networks, presented in the example in Section 3, aimed at minimizing the switching of individual blocks, allow for the formulation of an algorithm for the procedure, which includes the following steps:

1. Minimize the multi-output function.
2. Based on the output vectors obtained after minimization, determine the values of the Δ_{y_i} discriminants.
3. Create an output graph $G < Y, \vec{U} >$.
4. Sequential implementation of multi-output implicants associated with subsequent vertices of SOP networks.
5. Determine the SW value for SOP blocks and probabilities p(out), where p(out) denotes the probability of the appearance of the value 1 at the output of SOP blocks.
6. Create a switching graph describing the technology mapping of the multi-output function in the form of SOP networks.
7. Determine the value of the SW_{net} coefficient for the obtained mapping in the form of SOP networks.
8. Optimization of the SOP network, which comes down to the sequential implementation of one of the following three optimization methods:
   (i) Moving products from blocks closest to the inputs to blocks closest to the outputs of the SOP networks;
   (ii) Moving connections between blocks closest to the inputs to blocks closest to the outputs of the SOP networks;

   (iii) Splitting vertices and possibly reducing the number of SOP network nodes. If none of the optimization methods can be used, go to 10.

9. Determine the value of the SW_{net} coefficient for the new SOP networks. If the minimization of the value of the SW_{net} coefficient is achieved, go to 8.
10. End the optimization procedure.

5. Experimental Results

A series of experiments were conducted to confirm the effectiveness of the proposed methods of optimizing technology mapping. The experiments were carried out using a popular set of benchmarks described in the .pla [61] format. Each of the benchmarks considered is a description of a multi-output function f: $B^n \rightarrow B^m$. The experiments were performed by mathematical analysis using a spreadsheet. This type of approach leads to the need to limit the number of functions included in the set (m). We limited ourselves to examples where the number of m is three or four. This led to the need to extract three- or four-element sets from the original benchmarks.

The following series of experiments were conducted: classical implementation (Cl) [60]—as a reference value, implementation using a switching graph without optimization (SG), and implementation using a switching graph with the proposed optimization (SGopt). The experiments were conducted for three values of k (three, five, and seven), defining the number of terms in SOPs. For each considered solution, the SW coefficient was determined, the value of which directly affects the value of dynamic power losses. The obtained results are presented in Table 1.

Table 1. Experimental results—SW values.

	Benchmark			k = 3				k = 5				k = 7			
	Name	**In**	**Out**	**Cl**	**SG**	**SGopt**	**SG/SGopt**	**Cl**	**SG**	**Sgopt**	**SG/Sgopt**	**Cl**	**SG**	**Sgopt**	**SG/Sgopt**
1	bw_21_11_9	5	3	8.32	3.84	3.84	1.00	6.88	2.85	2.85	1.00	4.18	2.85	2.53	1.12
2	bw_27_26_25	5	3	4.00	1.99	1.99	1.00	3.71	1.89	1.89	1.00	2.91	1.89	1.34	1.41
3	bw_321	5	3	8.81	2.79	2.45	1.14	4.82	1.80	1.56	1.16	4.82	1.80	1.56	1.16
4	bw_432	5	3	7.84	2.89	2.65	1.09	4.03	1.90	1.66	1.15	3.87	1.90	1.66	1.15
5	ldd_mno	9	3	6.91	0.16	0.15	1.10	5.74	0.18	0.18	1.04	3.62	0.15	0.15	1.00
6	sao2_210	10	3	16.61	0.95	0.91	1.05	11.71	0.61	0.56	1.09	10.26	0.55	0.45	1.24
7	sao2_310	10	3	16.22	0.66	0.62	1.07	11.70	0.47	0.41	1.14	10.16	0.43	0.34	1.26
8	sao2_320	10	3	6.93	1.14	1.06	1.07	4.96	0.68	0.59	1.16	4.78	0.54	0.43	1.25
9	sao2_321	10	3	11.51	1.14	1.06	1.07	8.52	0.68	0.59	1.15	7.07	0.53	0.43	1.24
10	clip_432	10	3	28.25	5.42	5.36	1.01	19.59	3.13	2.94	1.07	16.44	2.43	2.12	1.15
11	rd53	5	3	10.99	9.62	9.52	1.01	7.20	7.06	6.73	1.05	6.31	4.97	4.84	1.03
12	rd73	7	3	30.37	28.70	28.48	1.01	20.73	16.26	15.67	1.04	14.99	12.75	11.85	1.08
13	sao2	10	3	17.09	3.12	2.94	1.06	12.30	2.25	2.21	1.02	10.76	2.07	1.81	1.14
14	sqn	7	3	15.02	4.83	4.32	1.12	9.83	3.06	2.91	1.05	8.86	2.51	2.33	1.08
15	Misex1_356	7	3	6.73	0.95	0.86	1.10	4.78	0.86	0.65	1.32	3.74	0.71	0.59	1.20
16	Misex1_367	7	3	6.30	0.96	0.93	1.03	4.42	0.87	0.75	1.16	3.49	0.87	0.84	1.04
17	Misex1_567	8	3	6.80	0.55	0.51	1.09	4.87	0.44	0.40	1.12	4.00	0.44	0.31	1.45
18	Misex1_3456	8	4	8.04	0.61	0.55	1.11	5.43	0.56	0.50	1.12	4.44	0.56	0.56	1.00
19	Misex1_3567	7	4	8.49	1.59	1.56	1.02	5.85	1.50	1.47	1.02	4.98	1.50	1.47	1.02
20	Misex1_4567	8	4	8.12	0.52	0.46	1.13	5.61	0.48	0.46	1.03	4.74	0.48	0.43	1.11
SUM:				233.35	72.46	70.23		162.67	47.54	44.98		134.42	39.94	36.04	
MEAN:							1.06				1.09				1.16

The first three columns of Table 1 contain information on the benchmarks under consideration. The benchmark name contains information on which single-output function of the multi-output function is considered. If the name does not contain this information, it means that all functions from the set are considered. For each case, information on the number of inputs (in) and the number of outputs (out) is provided.

The remaining columns of Table 1 are divided into three groups related to the values k = 3, 5, and 7. Each of these three groups contains four columns, which concern the following implementations of the system: classic implementation (Cl), mapping using a switching graph without optimization (SG), and mapping using a switching graph with the optimization presented in the article (SGopt). The individual cells contain the values of the obtained SW coefficients. The last column in each group with the SG/SGopt header contains the ratio of the switching activity values obtained using the SG and SGopt methods, respectively. The numbers placed there allow for a simple quantitative assessment of the optimization efficiency.

Table 1 has been supplemented with two additional rows. The first one (SUM) contains the total values of SW coefficients for Cl, SG, and SGopt. The second one (MEAN) describes the average value of SG/SGopt coefficients obtained for individual benchmarks. The values placed there allow for a synthetic presentation of the results in the form of bar charts shown in Figure 9.

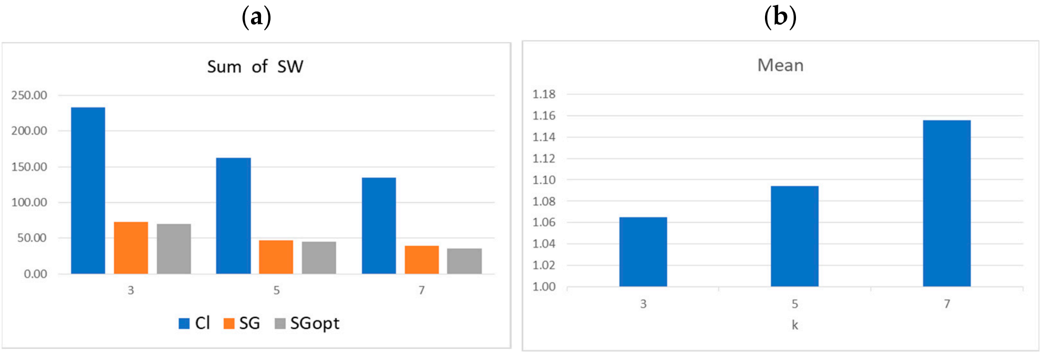


Figure 9. Synthetic presentation of the experimental results: comparison of the total SW value (**a**) and the average SW value (**b**).

The bar graph shown in Figure 9a shows a comparison of the total values of SW coefficients obtained for all analyzed benchmarks and three different SOP block sizes (k = 3, 5, 7). Analyzing the form of the graph, it can be seen that the methods using the switching graph in the mapping process significantly improve the efficiency of the obtained solutions in terms of the number of switches reflected in the SW value. There is no doubt that multi-level function implementations lead to a minimization of dynamic power losses. It can be seen that the optimization process leads to a further reduction in the SW value. It is worth emphasizing that the optimization leads to a reduction in the SW value for all considered SOP block sizes.

Figure 9b shows the average values of the comparison of the SG method with the SGopt method presented in the table. Values greater than one indicate that the optimization led to a reduction in the SW value and thus a reduction in the dynamic power. It is worth noting that with the increase in the k value, the optimization gains are greater and greater. It can therefore be seen that the optimization used leads to about a 10% reduction in the dynamic power.

6. Conclusions

The problem of designing energy-efficient digital systems is currently gaining increasing importance. It is becoming a key problem in the further development of cyber-physical systems, which commonly use the latest technological achievements. The area of searching for power consumption reduction is very wide and covers all stages of system synthesis. Potential gains can be achieved at all levels of system implementation, including low-level logic synthesis. The approach presented in the article presents the possibility of reducing dynamic power by searching for an appropriate technology mapping of the implemented systems. The process of searching for an energy-efficient solution is combined with the optimization of the proposed description of the system in the form of a switching graph. The switching graph is a bridge between the description of the technology mapping and its switching properties. It allows for directing the modification of the created SOP networks to minimize the switching of individual elements. It is worth emphasizing that the proposed optimization is universal and can be generalized to other architectures, not necessarily based on SOP.

The article focuses only on combinational circuits. The topic of reducing power consumption in sequential systems is not discussed. It is worth noting, however, that the appropriate implementation of FSMs combined with the energy-efficient coding of internal states in conjunction with the presented optimization methods can lead to extremely effective algorithms for the technology mapping of FSMs. This topic will be the subject of further work, which should lead to the development of methods for the synthesis of energy-efficient sequential circuits.

Author Contributions: Conceptualization, M.K. and D.K.; methodology, M.K. and D.K.; software, M.K.; validation, M.K. and D.K.; formal analysis, M.K. and D.K.; investigation, M.K. and D.K.; resources, M.K. and D.K.; data curation, M.K. and D.K.; writing—original draft preparation, M.K. and D.K.; writing—review and editing, M.K. and D.K.; visualization, M.K. and D.K.; supervision, M.K. and D.K.; project administration, M.K. and D.K.; funding acquisition, M.K. and D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kok, C.L.; Siek, L. A novel 2-terminal zener voltage reference. In Proceedings of the 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), Seoul, Republic of Korea, 7–10 August 2011; pp. 1–4. [CrossRef]
2. Kyaw, H.Y. Empowering Wearable Devices: Harnessing Ambient Energy for Sustainable Operation in IoT Systems. *Int. J. Eng. Sci. (IJES)* **2024**, *13*, 109–151.
3. Bondade, R. Hardware-software co-design of an embedded power management module with adaptive on-chip power processing schemes. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS), Paris, France, 30 May–2 June 2010; pp. 617–620.
4. Benini, L.; De Micheli, G. System-Level Power Optimization: Techniques and Tools. *ACM Trans. Des. Autom. Electron. Syst.* **2000**, *5*, 115–192. [CrossRef]
5. Hsiung, P.A.; Liu, C.W. Exploiting Hardware and Software Low power Techniques for Energy Efficient Co-Scheduling in Dynamically Reconfigurable Systems. In Proceedings of the International Conference on Field Programmable Logic and Applications, Amsterdam, The Netherlands, 27–29 August 2007; pp. 165–170.
6. Raghunathan, A.; Jha, N.K.; Dey, S. *High-Level Power Analysis and Optimization*; Springer: Boston, MA, USA, 1998; ISBN 978-1-4615-5433-2.
7. Ali, H.; Al-Hashimi, B.M. Architecture Level Power-Performance Tradeoffs for Pipelined Designs. In Proceedings of the IEEE International Symposium on Circuits and Systems, New Orleans, LA, USA, 27–30 May 2007; pp. 1791–1794.
8. Unsal, O.S.; Koren, I. System-level power-aware design techniques in real-time systems. *Proc. IEEE* **2003**, *91*, 1055–1069. [CrossRef]
9. Ratto, F.; Fanni, T.; Raffo, L.; Sau, C. Mutual Impact between Clock Gating and High Level Synthesis in Reconfigurable Hardware Accelerators, Eloectronics. *Electronics* **2021**, *10*, 73. [CrossRef]
10. Gagarski, K.; Petrov, M.; Moiseev, M.; Klotchkov, I. Power specification, simulation and verification of SystemC designs. In Proceedings of the 2016 IEEE East-West Design Test Symposium (EWDTS), Yerevan, Armenia, 14–17 October 2016; pp. 1–4.
11. Bard, S.; Rafla, N.I. Reducing power consumption in FPGAs by pipelining. In Proceedings of the 51st Midwest Symposium on Circuits and Systems, Knoxville, TN, USA, 10–13 August 2008; pp. 173–176.
12. Brooks, D.; Tiwari, V.; Martonosi, M. Wattch: A framework for architectural-level power analysis and optimizations. In Proceedings of the 27th International Symposium on Computer Architecture, Vancouver, BC, Canada, 14 June 2000; pp. 83–94.
13. Sunghwan, K.; Jihong, K. Low-power data representation. *Electron. Lett.* **2000**, *36*, 958–959.
14. Pursley, D.; Yeh, T. High-level low-power system design optimization. In Proceedings of the Symposium on VLSI Design, Automation and Test, Hsinchu, Taiwan, 24–27 April 2017; pp. 1–4.
15. Macko, D. Contribution to Automated Generating of System Power-Management Specification. In Proceedings of the 2018 IEEE 21st International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS), Budapest, Hungary, 25–27 April 2018; pp. 27–32.
16. Chen, C.; Srivastava, A.; Sarrafzadeh, M. On gate level power optimization using dual supply voltages. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2001**, *9*, 616–629. [CrossRef]
17. Chi, J.C.; Lee, H.H.; Tsai, S.H.; Chi, M.C. Gate Level Multiple Supply Voltage Assignment Algorithm for Power Optimization Under Timing Constraint. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2007**, *15*, 637–648. [CrossRef]

18. Manzak, A.; Chakrabarti, C. A low power scheduling scheme with resources operating at multiple voltages. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2002**, *10*, 6–14. [CrossRef]
19. Mukherjee, R.; Memik, S.O. Evaluation of Dual VDD Fabrics for Low Power FPGAs. In Proceedings of the Asia and South Pacific Design Automation Conference, Shanghai, China, 18–21 January 2005; Volume 2, pp. 1240–1243.
20. Shin, Y.; Seomum, J.; Choi, K.; Sakurai, T. Power gating: Circuits, design methodologies, and best practice for standard-cell VLSI designs. *ACM Trans. Design Autom. Electron. Syst.* **2010**, *15*, 1–37. [CrossRef]
21. Wang, N.; Zhong, W.; Chen, S.; Ma, Z.; Ling, X.; Zhu, Y. Power-gating-aware scheduling with effective hardware resources optimization. *Integr. Vlsi J.* **2018**, *61*, 167–177. [CrossRef]
22. Sain, M.; Shringi, S.; Asati, A. An Improved Power Gating Technique with Data Retention and Clock Gating. In Proceedings of the 2021 International Conference on Control, Automation, Power and Signal Processing (CAPS), 2021, Jabalpur, India, 10–12 December 2021. [CrossRef]
23. Raya, M.; Aviñó, O.; Busquets-Monge, S.; Perpiñá, X.; Vellvehi, M.; Jordà, X. Improvement of a self-powered gate driver power supply. In Proceedings of the 24th European Conference on Power Electronics and Applications (EPE'22 ECCE Europe), Hanover, Germany, 5–9 September 2022.
24. Fanni, T.; Sau, C.; Raffo, L.; Palumbo, F. Automated Power Gating Methodology for Dataflow-Based Reconfigurable Systems. In Proceedings of the 12th ACM International Conference on Computing Frontiers, Ischia, Italy, 18–21 May 2015.
25. Alam, M.R.; Nasab, M.E.S.; Fakhraie, S.M. Power Efficient High-Level Synthesis by Centralized and Fine-Grained Clock Gating. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2015**, *34*, 1954–1963. [CrossRef]
26. Bezati, E.; Casale-Brunet, S.; Mattavelli, M.; Janneck, J.W. Clock-Gating of Streaming Applications for Energy Efficient Implementations on FPGAs. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2017**, *36*, 699–703. [CrossRef]
27. Mullins, R.; Moore, S. Demystifying Data-Driven and Pausible Clocking Schemes. In Proceedings of the 13th IEEE International Symposiun on Asynchronous Circuits and Systems, Berkeley, CA, USA, 12–14 March 2007; pp. 175–185.
28. Park, Y.S.; Tao, Y.; Zhang, Z. A Fully Parallel Nonbinary LDPC Decoder with Fine-Grained Dynamic Clock Gating. *IEEE J. Solid-State Circuits* **2015**, *50*, 464–475. [CrossRef]
29. Xin, F.; Krstić, M.; Grass, E. Improvements of Pausible Clocking Scheme for High-Throughput and High-Reliability GALS Systems Design. In *Advanced Techniques in Logic Synthesis, Optimizations and Applications*; Springer: New York, NY, USA, 2011; pp. 401–417.
30. Chindhu, S.; Shanmugasundaram, N. Clock Gating Techniques: An Overview. In Proceedings of the Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, India, 2–3 March 2018. [CrossRef]
31. Özbaltan, M.; Berthier, N. Exercising Symbolic Discrete Control for Designing Low-power Hardware Circuits: An Application to Clock-gating. *IFAC-PapersOnLine* **2018**, *51*, 120–126. [CrossRef]
32. Shelar, R.S.; Narayanan, H.; Desai, M.P. Orthogonal partitioning and gated clock architecture for low power realization of FSMs. In Proceedings of the 13th Annual IEEE International ASIC/SOC Conference, Arlington, VA, USA, 13–16 September 2000; pp. 266–270.
33. Shen, W.; Cai, Y.; Hong, X.; Hu, J. Activity-Aware Registers Placement for Low Power Gated Clock Tree Construction. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI, Porto Alegre, Brazil, 9–11 March 2007; pp. 383–388.
34. Amini, E.; Najibi, M.; Pedram, H. A Novel Clock Generation Scheme for Globally Asynchronous Locally Synchronous Systems: An FPGA-Validated Approach. In Proceedings of the 15th Great Lakes Symposium on VLSI, Chicago, IL, USA, 17–19 April 2005; pp. 296–301.
35. Amini, E.; Najibi, M.; Pedram, H. Globally Asynchronous Locally Synchronous Wrapper Circuit Based on Clock Gating. In Proceedings of the IEEE Proceedings of Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures, Karlsruhe, Germany, 2–3 March 2006; pp. 193–199.
36. Krstic, M.; Grass, E.; Gurkaynak, F.K.; Vivet, P. Globally Asynchronous, Locally Synchronous Circuits: Overview and Outlook. *IEEE Des. Test Comput.* **2007**, *24*, 430–441. [CrossRef]
37. Kulmala, A.; Hamalainen, T.D.; Hannikainen, M. Reliable GALS Implementation of MPEG-4 Encoder with Mixed Clock FIFO on Standard FPGA. In Proceedings of the International Conference on Field Programmable Logic and Applications, Madrid, Spain, 28–30 August 2006; pp. 1–6.
38. Yu, Z.; Baas, B.M. High Performance, Energy Efficiency, and Scalability with GALS Chip Multiprocessors. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2009**, *17*, 66–79. [CrossRef]
39. Tang, B.; Bhave, S.; Manohar, R. Low Power Asynchronous VLSI with NEM Relays. In Proceedings of the 20th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), Potsdam, Germany, 12–14 May 2014.
40. Chen, Y.-H.; Chen, T.-C.; Tsai, C.-Y.; Tsai, S.-F.; Chen, L.-G. Algorithm and Architecture Design of Power-oriented H.264/AVC Baseline Profile Encoder for Portable Devices. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 1118–1128. [CrossRef]
41. Darabiha, A.; Chan Carusone, A.; Kschischang, F.R. Power Reduction Techniques for LDPC Decoders. *IEEE J. Solid-State Circuits* **2008**, *43*, 1835–1845. [CrossRef]
42. Park, Y.S.; Blaauw, D.; Sylvester, D.; Zhang, Z. Low-Power High-Throughput LDPC Decoder Using Non-Refresh Embedded DRAM. *IEEE J. Solid-State Circuits* **2014**, *49*, 783–794. [CrossRef]
43. Yong, G.; Yuan, Y.; Bei, F. The application of bus coding in designing of low power SoC based on Wishbone. In Proceedings of the 2012 International Conference on Computer Science and Information Processing (CSIP), Xi'an, China, 24–26 August 2012; pp. 1132–1135. [CrossRef]

44. Barkalov, A.; Titarenko, L.; Chmielewski, S. Improving characteristics of LUT-based Moore FSMs. *IEEE Access* **2020**, *8*, 155306–155318. [CrossRef]
45. Barkalov, A.; Titarenko, L.; Mielcarek, K. Improving characteristic of LUT-based Mealy FSMs. *Int. J. Appl. Math. Comput. Sci.* **2020**, *30*, 745–759. [CrossRef]
46. Li, S.; Choi, K. A high performance low power implementation scheme for FSM. In Proceedings of the 2014 International SoC Design Conference (ISOCC), Jeju, Republic of Korea, 3–6 November 2014. [CrossRef]
47. Chattopadhyay, S.; Yadav, P.; Singh, R. Multiplexer Targeted Finite State Machine Encoding for Area and Power Minimization. In Proceedings of the IEEE INDICON, India Annual Conference, 2004, Kharagpur, India, 20–22 December 2004; pp. 12–16.
48. Kajstura, K.; Kania, D. Low Power Synthesis of Finite State Machines State Assignment Decomposition Algorithm. *J. Circuits Syst. Comput.* **2018**, *27*, 1850041. [CrossRef]
49. Mengibar, L.; Entrena, L.; Lorenz, M.G.; Millan, E.S. Patitioned state encoding for low Power in FPGAs. *Electron. Lett.* **2005**, *41*, 948–949. [CrossRef]
50. Xia, Y.; Almaini, A.E.A. Genetic algorithm based state assignment for power and area optimization. *IEEE Proc. Comput. Digit. Tech.* **2002**, *149*, 128–133. [CrossRef]
51. Kubica, M.; Opara, A.; Kania, D. Logic Synthesis Strategy Oriented to Low Power Optimization. *Appl. Sci.* **2021**, *11*, 8797. [CrossRef]
52. Opara, A.; Kubica, M.; Kania, D. Decomposition Approaches for Power Reduction. *IEEE Access* **2023**, *11*, 29417–29429. [CrossRef]
53. Opara, A.; Kubica, M. Technology mapping of multi–output functions leading to the reduction of dynamic power consumption in FPGAS. *Int. J. Appl. Math. Comput. Sci.* **2023**, *33*, 267–284. [CrossRef]
54. Kubica, M.; Kania, D.; Kulisz, J. A Technology Mapping of FSMs based on Graph of Excitations and Outputs. *IEEE Access* **2019**, *7*, 16123–16131. [CrossRef]
55. Kubica, M.; Kania, D. Graph of Outputs in the Process of Synthesis Directed at CPLDs. *Mathematics* **2019**, *7*, 1771. [CrossRef]
56. Fanni, T.; Rodríguez, A.; Sau, C.; Suriano, L.; Palumbo, F.; Raffo, L.; de la Torre, E. Multi-Grain Reconfiguration for Advanced Adaptivity in Cyber-Physical Systems. In Proceedings of the 2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 3–5 December 2018; pp. 1–8.
57. Baumann, D.; Mager, F.; Singh, H.; Zimmerling, M.; Trimpe, S. Evaluating Low-Power Wireless Cyber-Physical Systems. In Proceedings of the 2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench), Porto, Portugal, 10–13 April 2018. [CrossRef]
58. Kubica, M.; Pochopień, B.; Kania, D. Switching activity reduction of SOP networks. *IEEE Access* **2024**, *12*, 112984–112994. [CrossRef]
59. Kania, D. A technology mapping algorithm for PAL-based devices using multi-output function graphs. In Proceedings of the 26-th Euromicro Conference, IEEE Computer Society Press, Maastricht, The Netherlands, 5–7 September 2000; pp. 146–153.
60. Kania, D. An Efficient Approach to Synthesis of Multi-Output Boolean Functions on PAL-based Devices. *IEEE Proc.-Comput. Digit. Tech.* **2003**, *150*, 143–149. [CrossRef]
61. Collection of Digital Design Benchmarks (Petr Fišer). 2008. Available online: https://ddd.fit.cvut.cz/www/prj/Benchmarks/ (accessed on 1 April 2016).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.