INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2016, VOL. 62, NO. 1, PP. 33-41
Manuscript received November 24, 2015; revised March, 2016. DOI: 10.1515/eletel-2016-0004

SMTBDD: New Form of BDD for Logic Synthesis

Marcin Kubica and Dariusz Kania

***Abstract*—The main purpose of the paper is to suggest a new form of BDD – SMTBDD diagram, methods of obtaining, and its basic features. The idea of using SMTBDD diagram in the process of logic synthesis dedicated to FPGA structures is presented. The creation of SMTBDD diagrams is the result of cutting BDD diagram which is the effect of multiple decomposition. The essence of a proposed decomposition method rests on the way of determining the number of necessary 'g' bounded functions on the basis of the content of a root table connected with an appropriate SMTBDD diagram. The article presents the methods of searching non-disjoint decomposition using SMTBDD diagrams. Besides, it analyzes the techniques of choosing cutting levels as far as effective technology mapping is concerned. The paper also discusses the results of the experiments which confirm the efficiency of the analyzed decomposition methods.**

***Keywords*—logic synthesis, SMTBDD, decomposition, technology mapping, FPGA, digital circuits**

I. Introduction

Dynamic development of Programmable Logic Devices, especially FPGA structures, requires using effective synthesis techniques. In the process of synthesis, dedicated to FPGA or CPLD structures, functional decomposition is particularly crucial as it is associated with the partition of a designed circuit into logic blocks which are included into a programmable structure [7, 8, 18, 19]. In most cases, circuits contain more than one output. A description of such circuits requires defining '*m*-element' output vectors in reply to '*n*-element' input vectors. Efficient decomposition of '*n*-variable' functions should be carried out for '*m*–multioutput functions' [6, 21]. In order to perform effective decomposition, an appropriate representation of Boolean function is especially vital. The representations of Boolean functions, which use Binary Decision Diagrams, turn out to be particularly useful [1, 4] as they have relatively small memory requirements reserved for allocating suitable data structures. The most popular type of BDD appears in the form of a reduced and organized diagram and is called ROBDD. It is required, for the representation of multioutput function, to use one of the most well-known types of BDD, i.e. SBDD (Shared BDD) (Fig.1b) [14, 15, 16, 23] or MTBDD (Multi Terminal BDD) (Fig. 1b) [13, 12, 15, 22].

Effective functional decomposition, dedicated to FPGA structures, impels to use alternative forms of representation BDD fragments. It can be observed in the case of multiple decomposition, for instance [6]. Multiroot diagram's parts, which include multivalue terminal nodes, are created as the result of multiple cutting. Thus, it is necessary to consider new kinds of BDD diagrams together with defining their properties and usefulness in the process of decomposition.

M. Kubica is with University of Bielsko – Biala (e-mail: mkinz@wp.pl).
D. Kania is with Silesian University of Technology (e-mail: dkania@polsl.pl).

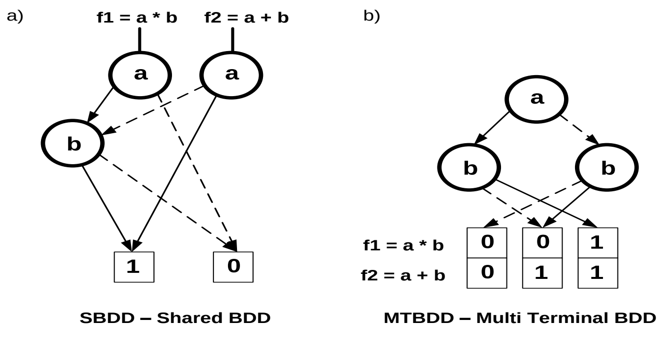


Fig. 1. The representation of multioutput function; a) SBDD, b) MTBDD

The main aim of the paper is to present an innovative type of purposes dedicated to FPGA structures. The proposed type of the diagram is a generalization of well-known diagram's kinds such as ROBDD, MTBDD, and SBDD. SMTBDD diagram which is created for decomposition

II. Theoretical Background

The first approach to the problem of decomposition was presented by Ashenhurst [2]. The issue was later developed by Curtis [5], Roth, and Karp [20].

Multioutput function f: $B^n \rightarrow B^m$ can be decomposed, i.e. $f(X_2,X_1) = F[g_1(X_1),g_2(X_1),\ldots g_p(X_1),X_2]$, where, X_1 (bound set), X_2 (free set) and $X_1 \cap X_2 = \Phi$ if, and only if, the column multiplicity $\upsilon(X_2|X_1)$ Karnaugh's map (partition matrix) satisfies the condition $\upsilon(X_2|X_1) \leq 2^p$.

The functions $g_1 \ldots\ldots g_p$ are sometimes called bound functions [5]. The partition of the circuit resulting from Curtis theorem is shown in Fig. 2. The $y_{m-1} \ldots\ldots y_0$ correspond to *m*-elements of multioutput function. A bound block is connected with a free block using '*p*' lines which relate to appropriate bound functions $g_1(X_1),g_2(X_1),\ldots g_p(X_1)$.

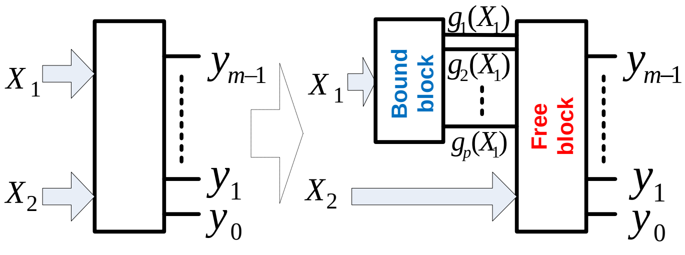


Fig. 2. Simple serial decomposition

In the case of BDD diagrams (ROBDD), a simple serial decomposition is associated with the cutting of a diagram [11]. BDD diagram is divided into two parts what directly relates to the partition of function's variables. One part of the diagram, occurring above the cutting line, is associated with a bound block. A lower part of the diagram is, on the other hand, associated with a free block. Cut nodes are situated below the cutting line and they are indicated on by edges emerging from the top part of the diagram. It turns out that the number of column patterns corresponds to the number of cut nodes. The following example of a function, which is described using a diagram in Fig. 3a, is worthy considering. As the result of a variable partitioning, *a* and *b* create a free set and variables *c*, *d*, and *e* form a bound set. The variable partitioning corresponds with the cutting of a diagram which was presented on Fig. 3a. In the diagram shown in Fig. 3a, there are two cut nodes that associate with '*b*' variable as the edges from the top part of the diagram (above the cutting line) are directed at these nodes. It means that a single bound function is adequate. The structure shown in Fig. 3b is the result of decomposition.

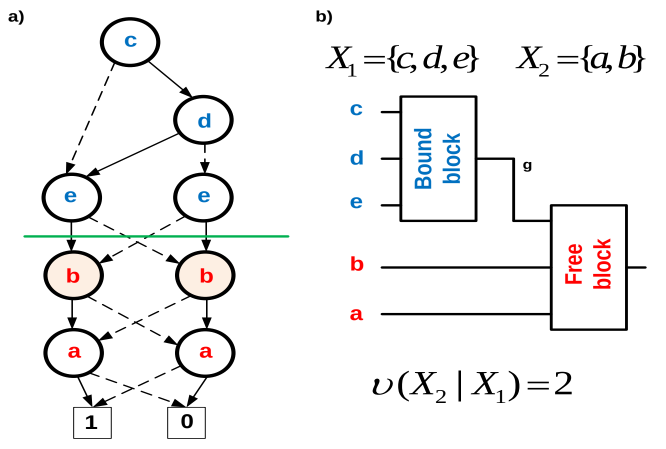

a)
b)
c
d
e
e
b
b
a
a
1
0
$X_1 = \{c, d, e\}$ $X_2 = \{a, b\}$
Bound block
g
Free block
$\upsilon(X_2 | X_1) = 2$

Fig. 3. A simple serial decomposition using BDD; a) description of a function, b) the result of decomposition

Apart from a simple serial decomposition, there are many various models of decomposition that are worthy mentioning such as non-disjoint decomposition, complex decomposition [5] (iterative and multiple), and mixed decomposition which is a combination of all the models above. Serial, non-disjoint decomposition is a generalization of a simple serial decomposition in which a single variable can be at the same time attached to both a free set as well as to a bound set (a variable replaces a bound function). Complex decompositions may turn out to be particularly useful if the number of function's arguments is considerably high and it is impossible for a decomposed circuit to be carried out on a single bound block and a single free block. As far as iterative decomposition is concerned, its main purpose is to divide a bound set in a sequential way which can lead to the growth of logic levels. An excessive number of logic levels makes dynamic properties of a designed circuit to get worsened. Multiple decomposition lacks this defect because each time a division of a free set is made. That is why, this form of decomposition is extremely vital and its core meaning is shown in Fig. 4. Multiple decomposition is possible in the situation when at least two simple serial decompositions exist and their bound sets are disjoint.

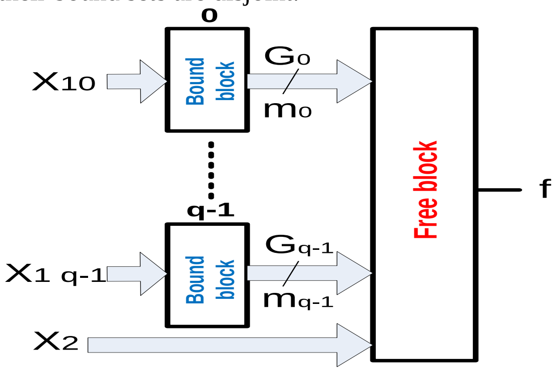


Fig. 4. Multiple decomposition

From the circuit's realization point of view, multiple decomposition turns out to be a very profitable model of function's decomposition. In the case of a function's description using BDD or MTBDD, it is connected with multiple cutting of a diagram which leads to a necessity of operating on new forms of diagrams, so called SMTBDD diagrams.

III. SMTBDD – New Form of BDD

SMTBDD diagram (Shared Multi Terminal Binary Decision Diagram) [9] is a binary decision diagram including '*n*' roots and '*p*' *m*-terminal nodes' bits ($p \leq 2^m$). The part of MTBDD graph placed between horizontal cutting lines (the top line – placed closer to the root and the bottom line – placed closer to the terminal nodes) is a good example of SMTBDD diagram. SBDD diagram is the most basic form of SMTBDD diagram and has two terminal nodes associated with 0 and 1 values.

The part of MTBDD (ROBDD) is the diagram separated by adjacent cutting lines or one cutting line for the part including the root. The parts of MTBDD (ROBDD) may appear in various forms of the diagrams such as BDD and SMTBDD in particular. The number of roots in SMTBDD diagram is equal to the number of cut nodes belonging to the analyzing part and indicated by the edges which come from the part above the cutting line. SMTBDD diagram may be dismantled into '*n*' MTBDD diagrams that have identical terminal nodes. The idea of splitting SMTBDD diagram into two MTBDD diagrams is presented on Fig.5.

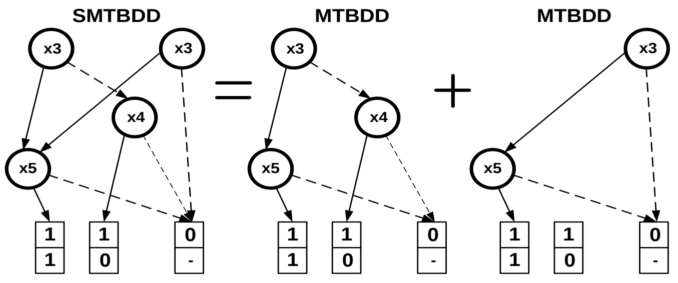


Fig. 5. SMTBDD as a symbolic connection of MTBDD

MTBDD is a one-root diagram. The process of multiple cutting of MTBDD diagram makes it possible to obtain MTBDD diagram and (at least one) SMTBDD diagram. Each BDD diagram is associated with a set of variables. The set of a diagram's variables (SMTBDD, ROBDD or

MTBDD), which is marked with 'E' symbol, is the set that corresponds with its nodes.

In order to illustrate the notions mentioned above, it is worth to analyze a simple example. Figure 6 clearly presents ROBDD diagram for which the set of a diagram's variables E={x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7, x_8}. In this case, the diagram is cut using three cutting lines. As the result of this process, three parts of the diagram are created. The part '0' includes ROBDD root and it is separated with the usage of only one cutting line. The cut part has just one root and three terminal nodes. In order to distinguish the terminal nodes from each other, it is necessary to use two bits. Thus, part '0' can be treated as MTBDD (SMTBDD with a single root) diagram for which the set of a diagram's variables E_0={x_0, x_1, x_2}. The part '1' creates a part of the diagram between two cutting lines. This diagram constitutes of three roots (the cut nodes indicated by the diagram from the part '0') and three terminal nodes where two bites are needed for them to be differentiated from each other. The part '1' is SMTBDD diagram that includes the nodes belonging to the set E_1={x_3,x_4,x_5}. The part '2' comprises a part of ROBDD between two consecutive cutting lines. The diagram, that is associated with this part, includes three roots and only two terminal nodes. Thus, it can be called SBDD (SMTBDD that has one-bit terminal nodes) diagram whose edges correspond with the set of variables E_2={x_6,x_7,x_8}. As the result of a multiple cutting of ROBDD diagram, shown in Fig. 6, three SMTBDD diagrams are created. Two of these diagrams especially worth mentioning, i.e. SMTBDD diagram including only one root (MTBDD) and SMTBDD diagram that has one-bit terminal nodes.

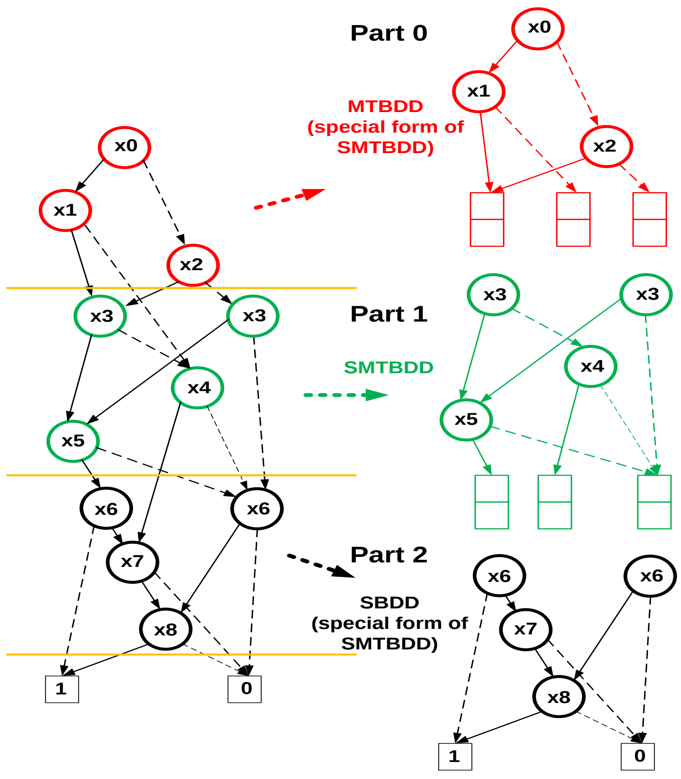


Fig. 6. The idea of obtaining SMTBDD from ROBDD diagram

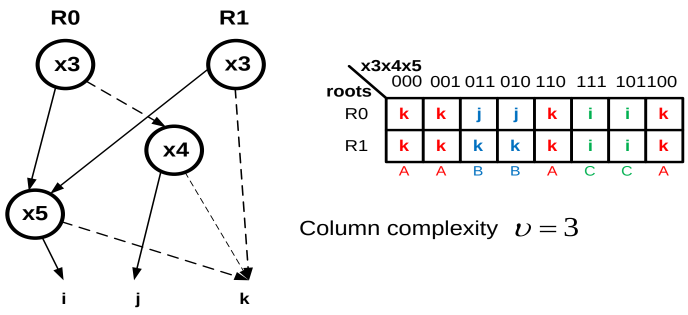
IV. Decomposition Using SMTBDD

In order to find appropriate decomposition, it is required to define the number of bound functions for each diagram's part (SMTBDD). Thus, it is necessary to introduce the notions of a root table and the complexity of a root table [10].

The root table of SMTBDD diagram is a two – dimensional table which columns are connected with suitable variables' combinations belonging to the set of 'E' variables of SMTBDD diagram and rows correspond with the roots of SMTBDD diagram. The cells of the root table are associated with nodes which are cut by 'the bottom' cutting line of the analyzed part. The vector of the cut node can be defined as the row of a root table connected with one SMTBDD root or MTBDD root.

The column multiplicity of the root table, marked with υ, is the number of various column's patterns of the root table associated with SMTBDD diagram.

Each SMTBDD diagram may be described using the root table. The columns of the root table are connected with SMTBDD paths. While choosing the root and analyzing the specific path in SMTBDD diagram, an appropriate terminal node 'is reached' and it is related to a suitable cut node. A similar situation can be observed in the case of the root table. While choosing an appropriate row (connected with the root) and a column (corresponding with the path), a proper cell may be unambiguously defined whose content is associated with the terminal node of SMTBDD diagram. The contents of proper columns create the patterns that may repeat. Thus, the column multiplicity is the number of various patterns of the columns of the root table. The idea of the column multiplicity is best illustrated on Fig. 7.



x3x4x5 / roots	000	001	011	010	110	111	101	100
R0	k	k	j	j	k	i	i	k
R1	k	k	k	k	k	i	i	k
	A	A	B	B	A	C	C	A

Column complexity $\upsilon = 3$

Fig. 7. SMTBDD diagram together with the root table - determining the column multiplicity

A newly proposed form of SMTBDD diagram is a 'multiroot' generalization of ROBDD or MTBDD diagrams. For the diagrams, which include only one root, the root table simplifies to one row (a vector of the cut nodes). Therefore, the column multiplicity of such a table is equal to the number of various cut nodes included in this vector.

The idea of determining the column multiplicity (the number of the cut nodes) for ROBDD diagram was presented on Fig. 8.

In order to define the column multiplicity, the following algorithm may turn out to be useful.

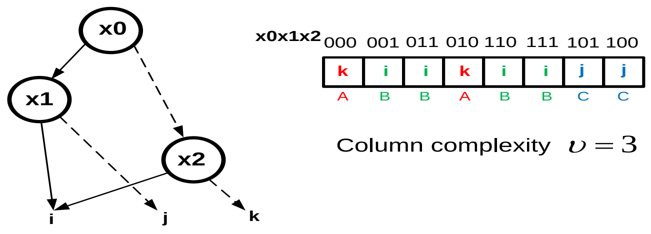


Fig. 8. ROBDD diagram together with a vector of the cut nodes – determining the column multiplicity

Algorithm 1 (creating the root table and determining the column multiplicity):
1. Formation of a vector of the cut nodes for each root by analyzing the paths
2. Placing a vector of the cut nodes in the root table in the row connected with a proper root
3. Assigning the number of various column patterns in the root table

The number of bound functions precisely depends on the number of column patterns or the number of various cut nodes for MTBDD (ROBDD) diagram. The idea of using SMTBDD diagrams in the process of decomposition is presented in the form of the following example.

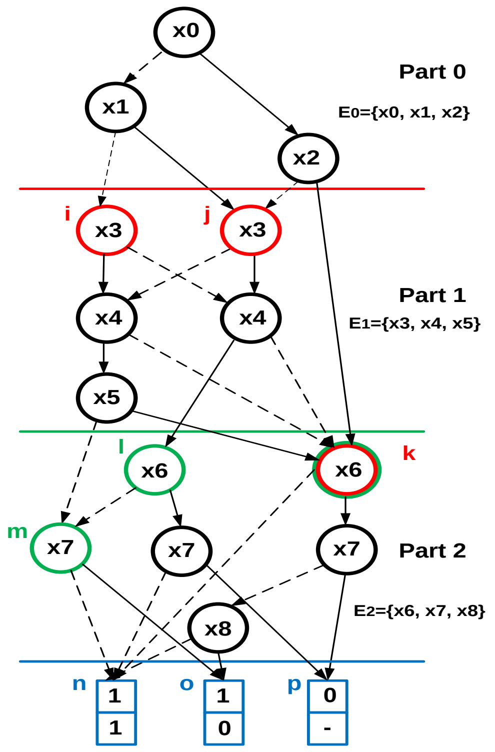


Fig. 9. MTBDD diagram describing multioutput function together with the cutting lines

Example 1
MTBDD diagram, which describes the multioutput function, is shown in Fig. 9. The diagram is divided into three parts.
In the case of the part 0, a single cutting line cut a diagram's part that 'indicates' three cut nodes. As the result of this cutting, the part presented on Fig. 10a, is created.

The part 0 is one – root MTBDD diagram. In order to define the number of '*g*' functions, it is necessary to create the vector of cut nodes and determine the number of various symbols included in this vector. The vector of the cut nodes is illustrated on Fig. 10a. It contains three various symbols associated with the nodes (i, j. k). It is essential to use two bits needed to distinguish three nodes what leads to decomposition in which two bound functions are obtained. After assigning codes to appropriate cut nodes, a reduced form of the part 0 is created. It contains the nodes corresponding with '*g*' functions which are shown in Fig. 10b.

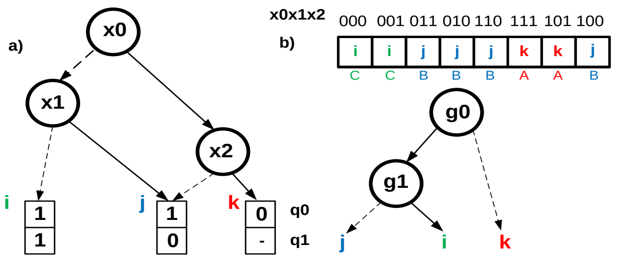
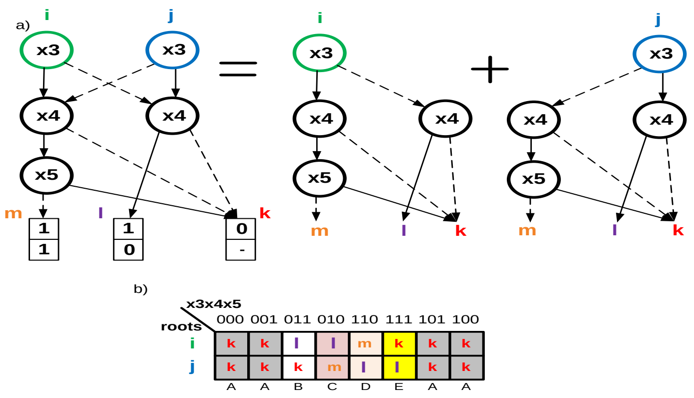


Fig. 10. Part 0; a) cut diagram's part, b) one – dimensional root table together with the part obtained after decomposition

It is necessary to analyze the part 1 after modification of the part 0. The analysis should start with describing the number of its roots. In the examined example, there are only two roots marked with 'i' and 'j' (Fig. 9) because the 'k' node, which is the cut node of the part 0, does not belong to the E1 set.
For the fact that SMTBDD diagram (the part 1) includes two roots, it is possible to divide it into two MTBDD diagrams what is symbolically presented on Fig. 11a. Next, in accordance with the 1 algorithm, the root table is formed. Particular rows of the table correspond with the roots of the part 1. As the result of this process, the root table, illustrated on Fig. 11b, is created (the table has two rows because SMTBDD diagram has two roots). Last but not least purpose of the 1 algorithm is to define the column multiplicity of the root table. The table shown in Fig. 11b contains five column patterns (A – E). Thus, it is a must to use three bits in order to distinguish them. In other words, there is a necessity to introduce the nodes standing for three bound functions. For the E1 set of variables also includes three elements, it is not worth transforming the part 1 introducing the nodes connected with bound functions.



roots	000	001	011	010	110	111	101	100
i	k	k	l	l	m	k	k	k
j	k	k	k	m	l	l	k	k
	A	A	B	C	D	E	A	A

Fig. 11. The part 1; a) SMTBDD diagram together with partition into two MTBDD diagrams, b) the root table

The analysis of the second part should begin with defining the number of its roots. In this part, the cut nodes may be found and they are indicated by the edges from the part 0 (k node) as well as from the part 1 (m, l, k nodes). The k node is a common cut node for two parts. Therefore, SMTBDD diagram has three roots (m, l, k) for the second part. SMTBDD diagram (the part 2) comprises the composition of three MTBDD diagrams (Fig. 12).

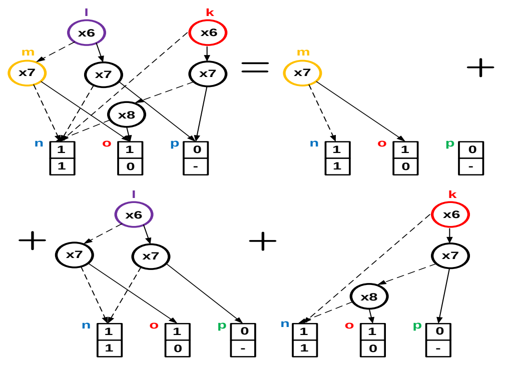
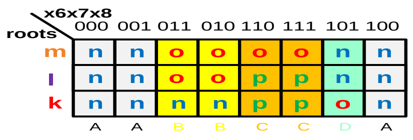


Fig. 12. SMTBDD diagram (the part 2) as the composition of three MTBDD diagrams

Similarly as in the case of the part 1, it is possible to decompose SMTBDD diagram into several MTBDD diagrams. Then, it is necessary to create the root table and determine the column multiplicity in accordance with the algorithm defining the column multiplicity. The root table, connected with the part 2, is presented on Fig. 13.



roots \ x6x7x8	000	001	011	010	110	111	101	100
m	n	n	o	o	o	o	n	n
l	n	n	o	o	p	p	n	n
k	n	n	n	n	p	p	o	n
	A	A	B	B	C	C	D	A

Fig. 13. The root table for the part 2

The root table (Fig. 13) includes four column patterns (A–D) that can be distinguished using two bits (two bound functions). The part 2 may be modified by substituting the nodes, associated with appropriate variables, with the nodes standing for bound functions. The final form of the diagram after transformations was shown in Fig. 14.

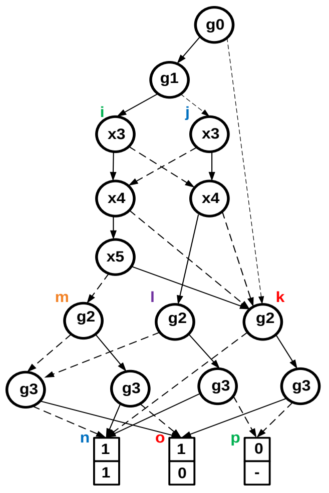


Fig. 14. MTBDD diagram describing multioutput function gained after multiple decomposition with the usage of the multiple cutting method

The whole procedure of decomposition should be carried out till the acquired free block is created in the LUT block that has a given number of inputs. As the result of multiple decomposition of the analyzed function, carried out using the multiple cutting method, the structure illustrated on Fig. 15 is obtained.

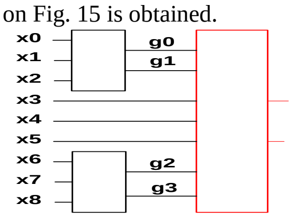


Fig. 15. First logic level (black colour) and a free block (red colour - for decomposition)

Carrying out the procedure of multiple cutting causes creating a successive logic level each time.

V. Non-Disjoint Decomposition Based on SMTBDD

Non-disjoint decomposition may be one of the ways of structure optimization in the process of technology mapping taking area into consideration. The essence of non-disjoint decomposition, which is generalization of a simple serial decomposition [22], is replacing some bound functions with variables *x* (inputs of a circle) that was shown in Fig. 16.

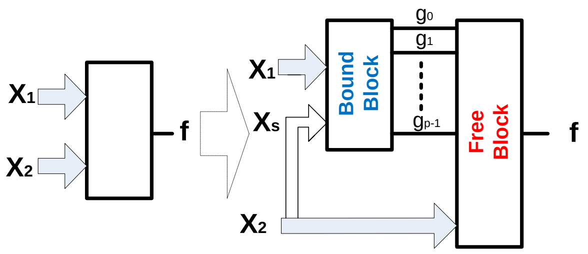
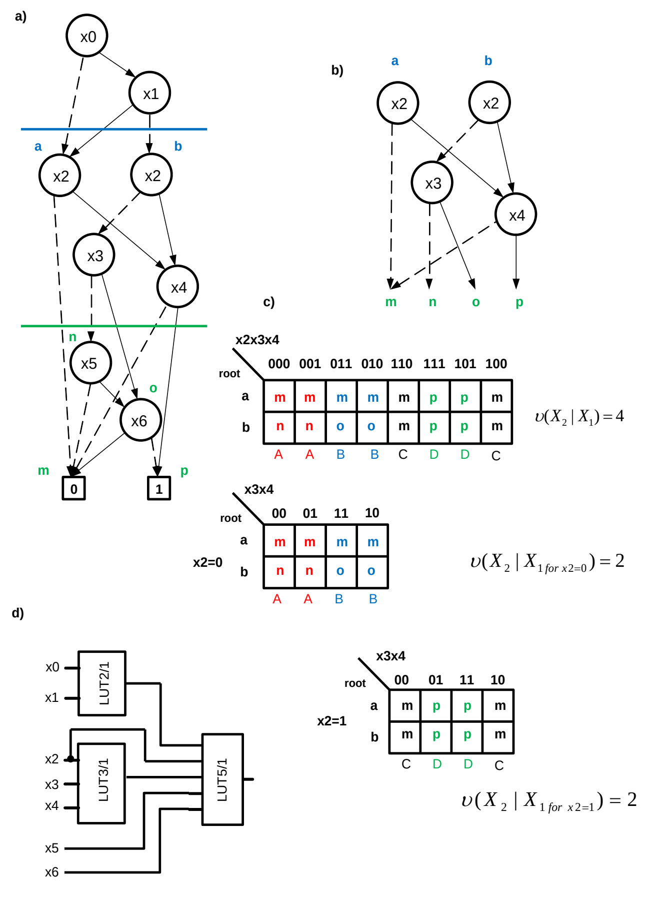


Fig. 16. The idea of non-disjoint decomposition

Creating a common set Xs may lead to the limitation of the number of blocks in a bound block. It is worth to mention that not each variable can fulfills a role of bound function. The process of searching non-disjoint decomposition should be started with searching disjoint decomposition for which separate variables are checked taking their profitability of attaching them to the set Xs into consideration. The idea of searching non-disjoint decomposition, for function description in the form of BDD, is presented in the papers [17, 26].

In the case of SMTBDD diagrams, the way of searching non-disjoint decomposition is very similar. It is crucial to define whether the attachment of a given variable to the set Xs is profitable. Each variable xi may take the value 0 ($xi = 0$) or 1($xi = 1$) that is connected with appropriate edges coming out from a given node. These edges indicate appropriate sub-diagrams. The following sub-diagrams $xi = 0$ and $xi = 1$ may be distinguished. Both of them indicate given number of cut nodes for given roots. There is a possibility to create root tables for $xi = 0$ and $xi =1$ for which column complexity may be described. The number of various column patterns determines the number of necessary bits (bound functions) for their differentiating in the case of the value of variables $xi= 0$ as well as $xi = 1$. When the number of bits (bound functions) necessary to distinguish column patterns of a root table for the nodes indicated by sub-diagram connected with $xi = 0$ is lower than the number of bits (bound functions) for disjoint decomposition and the number of bits (bound functions) for sub-diagram connected with *xi = 1* fulfills the same condition, then the variable xi may be bound function. Figure 17 illustrates the method of searching non-disjoint decomposition for multiroot SMTBDD diagram.



x2x3x4 / root	000	001	011	010	110	111	101	100
a	m	m	m	m	m	p	p	m
b	n	n	o	o	m	p	p	m
	A	A	B	B	C	D	D	C

$\upsilon(X_2 \mid X_1) = 4$

x2=0

x3x4 / root	00	01	11	10
a	m	m	m	m
b	n	n	o	o
	A	A	B	B

$\upsilon(X_2 \mid X_{1\,for\,x2=0}) = 2$

x2=1

x3x4 / root	00	01	11	10
a	m	p	p	m
b	m	p	p	m
	C	D	D	C

$\upsilon(X_2 \mid X_{1\,for\,x2=1}) = 2$

Fig. 17. Non-disjoint decomposition in SMTBDD diagrams a) ROBDD diagram together with cutting lines, b) SMTBDD diagram, c) root tables, d) the structure obtained

In Fig. 17 the usage of the variable x2 as 'switch over' variable is taken into consideration. For root tables associated with *x2 = 0* and *x2 = 1*, column complexity is 2. Thus, in both cases a single bit is needed in order to distinguish column patterns (single bound function). In both cases, *x2 = 0* as well as *x2 = 1*, the number of bound functions is lower than the number of bound functions for disjoint decomposition (where two bound functions are needed). It means that variable *x2* may fulfill the role of 'switch over' function.

VI. Technology Mapping Based on SMTBDD

Logic blocks, included in modern FPGA circuits, are characterized by substantial flexibility. It enables to configure the number of inputs of a configurable logic block in such a way to obtain the best technology mapping as possible. Naturally, configurable possibilities of blocks are limited and the number of inputs of a single logic block is not higher than 7. It is worthy mentioning that the number of inputs of a logic blocks plays a crucial role in the case of decomposition carried out using multiple cutting method as it is almost equal to the number of variables in the set E. The essence of technology mapping is choosing the cutting line of SMTBDD diagram that will best reflect configurable possibilities of a logic block and guarantee at the same time the highest reduction of the number of variables in SMTBDD. Figure 18 presents the idea of choosing the level of a cutting line in the process of technology mapping.

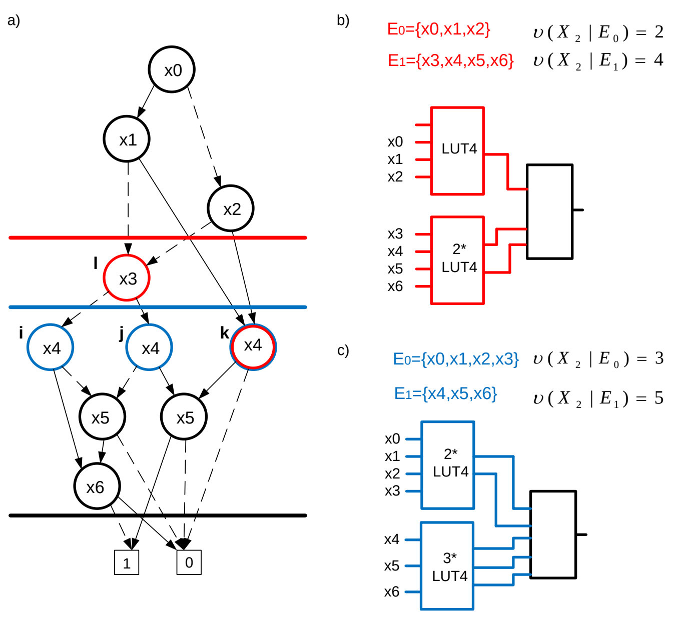


Fig. 18. The idea of choosing the cutting line in SMTBDD

Figure 18a illustrates two alternative methods of cutting SMTBDD dedicated to technology mapping to the blocks that has four inputs and one output (LUT4/1). In the first case, the set E0 has three elements and the set E1 has four elements. It leads to the necessity of carrying out three bound functions in a bound block (three blocks LUT4/1 are necessary). In the second case, the set E0 has four elements while the set E1 has three elements. It leads to the necessity of carrying out five bound functions (five blocks LUT4/1 are necessary). Thus, it can be said that appropriate choosing of a cutting line may lead to considerably better technology mapping by limitation of logic blocks. It is crucial to define the way of choosing the cutting line. It enables to introduce efficiency factor of technology mapping δ (1):

$$\delta = \text{numb\_of\_blocks} - (\text{card}(E) - \text{numb\_of\_g}) \qquad (1)$$

The value of efficiency factor of mapping δ depends on two parameters *card(E)* and *numb_of_g*. They are the result of a complex way of the division of arguments (*card(E))* and the analysis of decomposition parameters obtained (*numb_of_g)*. One more parameter called the *numb_of_blocks* can be distinguished and it defines the number of necessary logic blocks while carrying out. The essence of searching best possible cutting lines is the analysis of the value of the factor δ For various cutting levels resulting from configurable abilities of logic blocks considered. The best technology mapping may be obtained for the cutting levels in which the value of the factor δ is the lowest.

VII. Experimental Results

Choosing the cutting line depends on configurable abilities of the available logic blocks. Thus, in order to check the effectiveness of synthesis methods using SMTBDD, it is necessary to introduce hypothetical LUT blocks that have k inputs and one output (LUTk/1) and which can, but not necessarily, use all of the k inputs after the implementation of decomposed function. The analysis of technology mapping was conducted for chosen benchmarks [3] for LUT blocks in which the number of inputs (k) takes the values ranging from 4 to 8 (blocks that have higher number of inputs are not used taking substantial silicon area into consideration). Assuming that separate sets of variables E may include not more than k elements, it can be defined which configurations (what is the number of inputs) were used the most often for a given maximum k. the results are presented in Table 1.

Table 1. The results of synthesis for LUTk/1 blocks

Bench.	input	output	max k = 5		max k = 6			max k = 7				max k = 8				
			Lut 4/1	Lut 5/1	Lot 4/1	Lut 5/1	Lut 6/1	Lut 4/1	Lut 5/1	Lut 6/1	Lut 7/1	Lut 4/1	Lut 5/1	Lut 6/1	Lut 7/1	Lut 8/1
9sym	9	1	9	0	0	0	4	0	1	0	3	1	0	0	0	2
rd73	7	3	6	1	3	0	4	0	0	0	3	0	0	0	3	0
rd84	8	4	10	1	3	1	4	4	0	0	5	0	0	0	0	4
5xp1	7	10	7	6	7	1	4	3	1	1	4	3	1	1	4	0
z5xp1	7	10	11	3	7	1	4	3	1	1	4	3	1	1	4	0
con1	7	2	1	2	0	1	1	0	1	1	0	0	1	1	0	0
sqr6	6	11	12	6	3	3	4	3	3	4	0	3	3	4	0	0
inc	7	9	5	17	3	1	9	1	1	5	2	1	1	5	2	0
misex1	8	7	7	7	4	0	7	2	0	3	2	2	0	3	2	0
sqn	7	3	1	10	3	0	6	0	0	0	3	0	0	0	3	0
f51m	8	8	6	4	5	1	3	4	1	1	2	3	1	1	1	1
b12	15	9	7	13	5	4	9	4	2	1	6	2	2	1	2	3
z4ml	7	4	6	3	3	1	1	1	1	0	2	1	1	0	2	0
x2	10	7	8	7	6	0	6	4	2	1	3	6	0	1	0	3
clip	9	5	17	7	1	0	13	0	0	2	7	5	0	0	0	8
misex2	25	18	21	20	11	3	21	9	5	2	12	3	5	1	2	11
Sum:			**134**	**107**	**64**	**17**	**100**	**38**	**19**	**22**	**58**	**33**	**16**	**19**	**25**	**32**

Figure 19 presents the diagrams illustrating the usage of separate configurations for given values max k. While analyzing the graphs, it may be said that the most often using configurations are those for which the number of inputs is equal to max k and it should lead to good dynamic properties of the structures obtained. It is also worth to mention that LUT4/1 is the most often used configuration which can be connected with appearing the function, in the analyzed benchmarks, that has relatively small number of inputs and does not require decomposition.

The obtained results indicate that cutting levels, which are far from each other at max k variables in SMTBDD, plays a key role for good technology mapping. Besides, good results may be also gained by searching decompositions associated with the cutting lines for which E=4. These observations may have considerable influence on searching effective mappings in relatively short time.

Effectiveness of the synthesis method, based on SMTBDD diagram, may be assessed by comparing the results of decomposition carried out with the usage of commercial tools offered by the producers of FPGAs. There is a possibility to liken the results of synthesis only partly conducted in

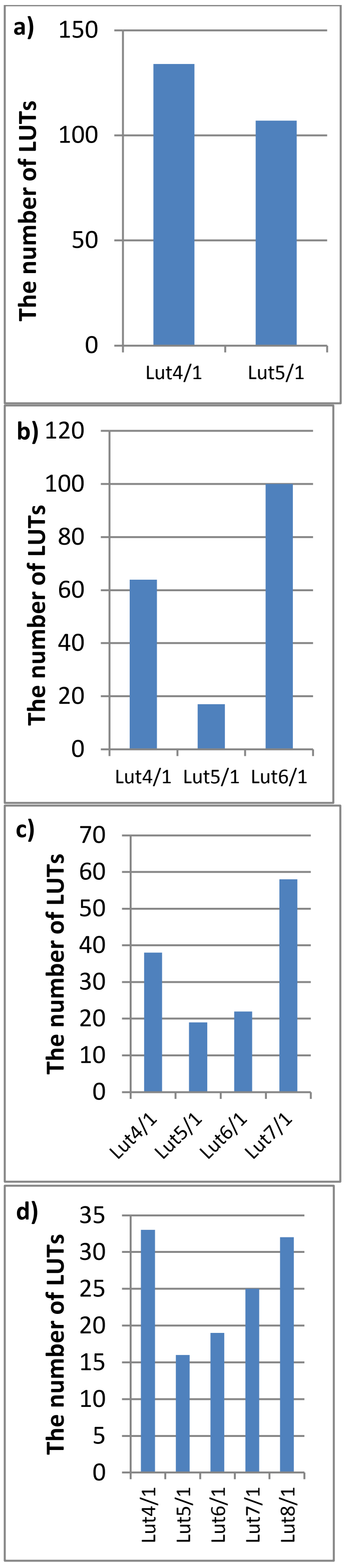


Fig. 19. The number of LUT blocks used that have determined number of inputs; a) for the blocks having max 5 inputs, b) for the blocks having max 6 inputs, c) for the blocks having max 7 inputs, d) for the blocks having max 8 inputs

commercial tools (conducting some stages associated with decomposition, based on SMTBDD diagram, is done beyond the tool delivered by the producer) with the results gained for the synthesis carried out entirely in a commercial tool. In order to make the cooperation with the commercial tool possible, a description of a decomposed circuit in Verilog HDL is created.
The first table best presents the results of the synthesis gained by using various methods. The previously mentioned synthesis was conducted in ISE 14.7 tool [25] for Spartan XC3S200 circuits [24] in which the blocks LUT 4/2 or LUT 5/1 are placed. The number of logic blocks is included in the column named LB. These blocks have a proper number of single LUT 4/1 blocks (L4 column). It is worth to mention that two LUT 4/1 blocks are treated as one LUT 5/1 block.

The first three columns of the table 2 include the name of the benchmark, the number of inputs and the number of outputs. The column named '*ISE*' comprises the results of the synthesis in which all the synthesis stages are conducted using ISE tool. The following two columns ('*SMTBDD (equation description) + ISE'* and '*SMTBDD (table description) + ISE*' include the results of the synthesis in which the initial synthesis stages were carried out on the basis of SMTBDD diagrams and the outcome was presented in Verilog HDL. A difference between the columns '*SMTBDD (equation description) + ISE' and 'SMTBDD (table description) + ISE'* depends on the form of the circuit description after decomposition. A decomposed and initially defined circuit on the Boolean level may take two forms. The first one can appear as a table description of particular modules that are connected with *k* – inputs of LUT blocks. The second one may take a form of a description using logic equations that include the number of variables, not higher than *k*. It seems that a description of separate modules, using logic equations, gives more freedom to the commercial tool in the process of optimization conducted in the ISE system, in addition.

Table 2. The results of synthesis for Spartan 3 circuit carried out using ISE tool

			ISE		**SMTBDD (equation description) + ISE**		**SMTBDD (table description) + ISE**	
Bench.	**in**	**out**	**LB**	**L4**	**LB**	**L4**	**LB**	**L4**
9sym	9	1	8	15	5	9	5	9
t481	16	1	3	5	6	11	6	11
rd73	7	3	9	17	4	8	4	8
rd84	8	4	13	21	7	12	6	12
5xp1	7	10	12	23	10	20	10	20
z5xp1	7	10	13	22	9	17	9	18
con1	7	2	3	5	3	5	3	5
sqr6	6	11	13	24	12	23	12	24
misex1	8	7	9	16	10	18	11	21
sqn	7	3	10	19	10	19	11	21
f51m	8	8	13	26	7	15	7	15
b12	15	9	15	26	17	30	17	33
z4ml	7	4	3	6	5	9	6	12
x2	10	7	8	15	8	15	11	22
clip	9	5	25	46	17	31	17	31
misex2	25	18	21	36	30	48	32	61
SUM:			**178**	**322**	**160**	**290**	**167**	**323**

The acquired results indicate that using SMTBDD diagrams on particular synthesis stages leads to the improvement of the overall outcome. The total number of blocks, which was obtained as the result of the synthesis of the circuit's table description and the description using circuits' tested equations (similar results in SMTBDD columns (table description) + ISE, SMTBDD (equation description) + ISE, SMTBDD), is lower than the total number of the blocks gained as the result of the synthesis conducted exclusively in the ISE system. Apart from that, it is clearly presented that the description of a decomposed circuit in Verilog HDL, appearing in the form of equations, gives considerably better results. In the case of defining the circuits using equations, the ISE system probably exploits additional optimization possibilities which are connected with the specificity of the circuits used. There are slight differences that may result from the way of summing the blocks' 'halves'. Thus, it may have an influence on the opportunity of an impartial comparison of the efficiency, as far as the analyzed methods are concerned, with the commercial tools.

VIII. Conclusion

One of the techniques, which enables to limit the number of logic levels (improvement in the dynamic properties of the circuit), is multiple decomposition. It may be carried out in the process of multiple cutting of BDD diagram in which it is not necessary to change the set of variables. This feature is considered to be its main advantage. While cutting BDD diagram, using more than one cutting line, parts are created. These parts form a new type of diagrams, so called SMTBDD diagrams. In order to conduct this decomposition, it was essential to devise the methods that are necessary to define the number of bound functions. That is why, the notions of the root table and the column multiplicity of the root table were introduced. The results of the experiments indicate that using decomposition based on multiple decomposition (carried out with the usage of SMTBDD diagrams) often leads to the reduction of the number of the blocks used (especially in the description using equations). This conclusion is an incredibly precious remark, especially in the case of further research papers in the field of synthesis. One of the crucial aspects is using decomposition techniques and direct them to the problems of reducing the power consumption of acquired structures.

References

[1] S. B. Akers, "Binary Decision Diagrams", *IEEE Transactions on Computers*, Vol. C-27, No.6, June 1978, pp.509-516
[2] R. L. Ashenhurst, "The Decomposition od switching functions, *Proceedings of an International Symposium on the Theory of Switching*, 1957, pp.74-116
[3] Benchmarking And Experimental Algorythmics Laboratory, A Benchmark set, 2008, http://www.cbl.ncsu.edu:16080/ benchmarks/ LGSynth93/testcase/
[4] R. E. Bryant, "Graph Based Algorithms for Boolean Function Manipulation", *IEEE Transactions on Computers* vol.C-35, no. 8, 1986, pp. 677-691
[5] H. A. Curtis, H.A., "A New Approach to the Design of Switching Circuits", D. van Nostrand Company Inc, New York, 1962
[6] D. Kania, „Układy Logiki Programowalnej", Wydawnictwo Naukowe PWN, Warszawa, 2012
[7] D. Kania and J. Kulisz, "Logic synthesis for PAL-based CPLD-s based on two-stage decomposition", *The Journal of Systems and Software*, vol. 80, 2007, pp. 1129-1141

[8] D. Kania and A. Milik, "Logic Synthesis based on decomposition for CPLDs", *Microprocessor and Microsystems*, vol. 34, 2010, pp. 25–38

[9] M. Kubica and D. Kania, „Dekompozycja wielokrotna z wykorzystaniem SMTBDD", *Elektronika; konstrukcje, technologie, zastosowania,* 11, 2013, pp. 83-87

[10] M. Kubica and D. Kania, "SMTBDD: New Concept of Graph for Function Decomposition", *13th IFAC and IEEE Conference on Programmable Devices and Embedded Systems, PDES 2015, The proc. of PDES 2015*, Vol. 48, Issue 4, 13-15 May 2015, Cracow, pp. 49–54

[11] Ch. Legl, B. Wurth, nad K. Eckl, "A Boolean Approach to Performance – Directed Technology Mapping for LUT – Based FPGA Designs", *33th Design Automation Conference*, 1996, pp. 730-733

[12] P. Mikusek and V. Dvorak, "Heuristic Synthesis of Multi – Terminal BDDs Based on Local Width/Cost Minimization", 12th Euromicro Conference on Digital System Design / Architectures, Methods and Tools, 2009, pp. 605-608

[13] P. Mikusek, "Multi – Terminal BDD Synthesis and Applications", *Field Programmable Logic and Applications*, 2009, pp. 721-722

[14] S. Minato, N. Ishiura, and S.Yajima, "Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation", *27th ACM/IEEE Design Automation Conference*, 1990, pp. 52-57

[15] S. Minato, "Binary Decision Diagrams and Applications for VLSI CAD", Kluwer Academic Publishers, 1996

[16] H. Ochi, N. Ishiura, and S.Yajima, "Breadth – First Manipulation of SBDD of Boolean Functions for Vector Processing", *28th ACM/IEEE Design Automation Conference,* 1991, pp. 413-416

[17] A. Opara, „Dekompozycyjne metody syntezy układów kombinacyjnych wykorzystujących binarne diagramy decyzyjne", Rozprawa doktorska, Instytut Informatyki, Politechnika Śląska, Gliwice 2008

[18] A. Opara and D. Kania, "Decomposition-based Logic Synthesis for PAL-based CPLDs", *International Journal of Applied Mathematics and Computer Science (AMCS),* Vol. 20, No. 2, 2010, pp. 367-384

[19] M. Rawski, L. Jóźwiak, M. Nowicka, and T. Łuba, "Non – Disjoint Decomposition of Boolean Functions and Its Application in FPGA – oriented Technology Mapping", *Proceedings od the 23rd EUROMICRO Conference*, 1997, pp. 24 - 30

[20] J. P. Roth and R. M. Karp, "Minimization Over Boolean Graphs", *IBM Journal of Research and Development*, 1962, pp. 227-238

[21] Ch. Scholl, "Functional Decomposition with Application to FPGA Synthesis", Kluwer Academic Publisher, Boston, 2001

[22] Ch. Scholl, B. Backer, and A. Brogle,: The Multiple Variable Order Problem for Binary Decision Diagrams: Theory and Practical Application,: *Design Automation Conference, Proceedings of the ASP-DAC'01,* 2001, pp. 85 - 90

[23] M. A. Thorton, J. P. Williams, R. Drechsler, N. Drechsler, D. M. Wessels, "SBDD Variable Reordering Based on Probabilistic and Evolutionary Algorithms", *Communications, Computers and Signal Processing*, 1999, pp. 381-387

[24] Xilinx, Spartan-3 Generation FPGA User Guide (UG331), 2011

[25] Xilinx, *ISE Design Suite 14*, UG631, 2013

[26] S.Yamashita, H.Sawada, and A.Nagoya, "New Methods to Find Optimal Non – Disjoint Bi – Decompositions", *Design Automation Conference, Proceedings of the ASP-DAC '98*, 1998, pp. 59 - 68