Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-6 (2018) 107–112

Technology mapping of multi-output function into LUT-based FPGA

Marcin Kubica*, Adam Milik, Dariusz Kania*****

* University of Bielsko - Biala, Bielsko - Biala, Poland (e-mail: mkinz@wp.pl)
** Silesian University of Technology, Gliwice, Poland (e-mail: adam.milik@polsl.pl)
*** Silesian University of Technology, Gliwice, Poland (e-mail: dkania@polsl.pl)

Abstract: The paper presents the essence of technology mapping of multi-output functions in modern logic blocks of FPGA circuits. Impressive configurable abilities of the blocks enable to lead decomposition in a way to use available logic resources of LUT-based FPGAs. As modern configurable logic blocks include several LUT blocks, it is necessary to search for such a decomposition model that would enable technology mapping simultaneously in LUT blocks and in whole configurable logic blocks. In the paper, the authors present a series of experiments and prove the effectiveness of multiple decomposition and presented methods.

Keywords: Logic synthesis, SMTBDD, Decomposition, FPGA, Combinatorial circuits.

1. INTRODUCTION

Substantial increase of complexity and flexibility of modern FPGA circuits gave the beginning of efficient synthesis tools. The key element of synthesis is decomposition of multi-output function that is the theoretical basis of a circuit's partitioning into logic blocks available in FPGA. Decomposition process is inextricably from technology mapping that enables to use configurable abilities of blocks in a synthesis process. In the case of combinational circuits, configurability of a logic block is connected with a minor possibility of changing number of its inputs (*k*). Logic block is able to implement a given logic function that has a limited (usually low) number of arguments.

Decomposition theory is based on the theorem by Ashrenhurst – Curtis (Ashenhurst, 1957; Curtis, 1962), that describes the partition of variables into a bound set *Xb* and a free set *Xf*. In the process of technology mapping, appropriate sets are associated with the functions carried out in a bound and a free block, respectively. Connections between these blocks correspond with so called bound functions whose number is determined in the theorem (Ashenhurst, 1957; Curtis, 1962). The model of partition, described in the theorem, enabled to create a series of decomposition tools. Originally, such tools as MIS-PGA (Murgai et al., 1991), ASYL (Abouzeid et al., 1993) or Chortle (Francis et al., 1990) are dedicated to gate structures synthesis and delivers solutions far from optimum. Newer tools such as BDS – PGA 2.0 (Vemuri et al., 2002), IRMA2FPGA (Jozwiak and Chojnacki, 2003) or dekBDD (Opara and Kania, 2010) give much better solutions taking the number of used logic blocks into account as well as the number of logic levels (a delay driven decomposition).

ABC system (Berkeley Logic Synthesis Group) should be especially concerned as it is dedicated to resynthesis process (Fiser and Schmidt, 2009; 2012).

From the point of view of carrying out decomposition, the key element is the way of logic function representation. Especially effective way of function representation (taking into account its memory usage and the speed of executing operations) turns out to be binary decision diagrams (BDD) (Bryant, 1986). Basically, its ordered and reduced form is used, i.e. ROBDD. For function representation, modified forms of BDD may be used such as SBDD (Minato, 1996), MTBDD (Minato, 1996) or SMTBDD (Babu and Sasao, 1998; Kubica and Kania, 2015).

The goal of the paper is to present theoretical basis and experiments enabling for efficient mapping of multi-output function into LUT-based FPGAs. Technology mapping process is dedicated at logic blocks that are core of modern FPGA circuits.

2. THEORETICAL BACKGROUND

In the case of function representation in the form of BDD, the essence of carrying out decomposition is making a horizontal cutting of a diagram shown in figure 1.

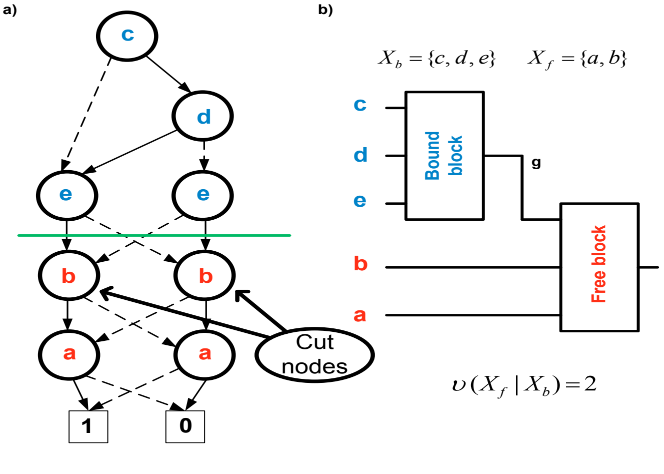
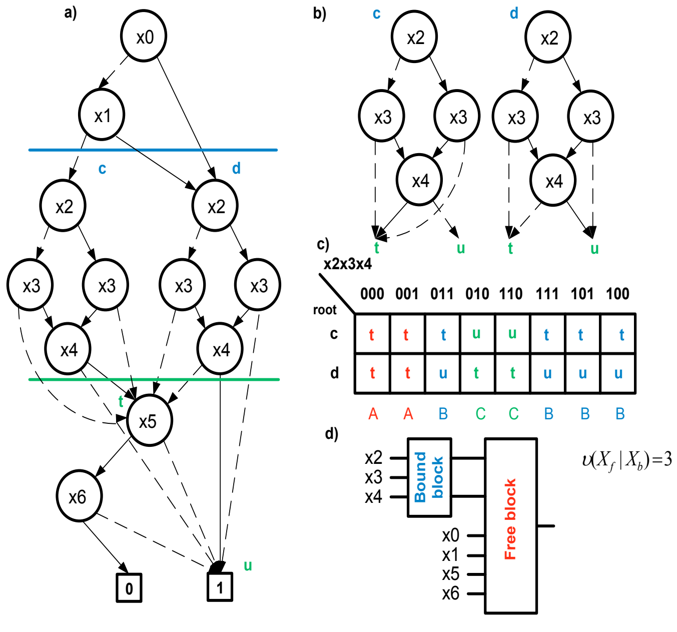


Fig. 1. Decomposition by Ashrenhurt – Curtis, a) ROBDD, b) Gained partition of a circuit

2405-8963 © 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.07.138

Figure 1 illustrates a single root node ROBDD diagram that underwent a horizontal cutting. The result is the partition of the diagram into two parts. The top part is associated with a bound set *Xb = {c,d,e}*. The bottom part is associated with a free set *Xf*. On figure 1a, cut nodes were indicated. The nodes are placed in the bottom part of the diagram and they are pointed by the edges from the top part. The number of cut nodes *υ(Xb|Xf)* is 2. Thus, in order to distinguish them a single bound function g is needed. A gained partition of a circuit is presented on figure 1b.

Obviously, decomposition may be also undergone by the functions described using multi-root diagrams such as SMTBDD (Kubica and Kania, 2017a; 2017b). In general, each time the number of cuttings of BDD diagram may be higher than one. In this situation, neighboring cutting lines divide a diagram into extracts connected with several bound sets. The idea of carrying out decomposition by introducing several cutting lines is illustrated on figure 2.



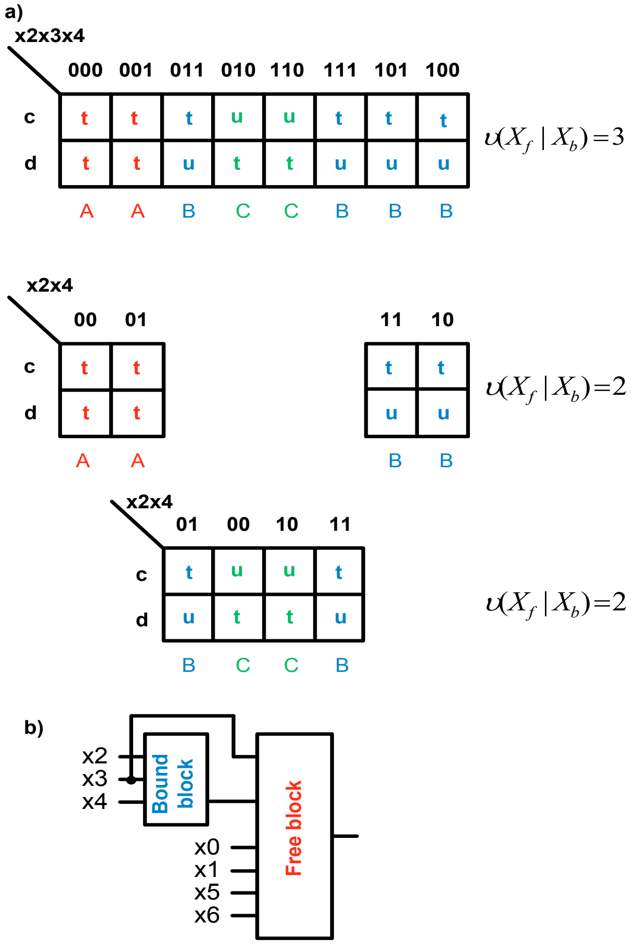
x2x3x4 root	000	001	011	010	110	111	101	100
c	t	t	t	u	u	t	t	t
d	t	t	u	t	t	u	u	u
	A	A	B	C	C	B	B	B

Fig. 2. Decomposition carried out using the method of multiple cutting, a) ROBDD, b) the extract of a diagram placed between cutting lines, c) a root table, d) a gained partition

The diagram on figure 2a presents two cutting lines. The extract placed between cutting lines and associated with the variables *{x2,x3,x4}* is a bound set *Xb*. The extract is illustrated on figure 2b. The diagram may be treated as a single SMTBDD (Kubica and Kania, 2015) diagram that has to roots c and d. In order to indicate the number of needed bound functions, it is necessary to create a root table connected with this extract hat is shown in figure 2c. Separate rows of a root table correspond with the roots of SMTBDD. In cells of a table are placed the symbols connected with cut nodes that are obtained for a path in a diagram associated with a given column. Column complexity of a root table *υ(Xb|Xf)* is the number of various column patterns in a root table. It determines the number of needed bound functions. Figure 2c presents three column patterns marked as A, B, and C. Figure 2d illustrates the structure obtained as the result of decomposition.

In the analyzed case, the extract placed above the top cutting line may be associated with a separate bound block. The place of a bound set consisted of the variables *x0* and *x1* may be taken by a single bound function as in order to distinguish between cut nodes c and d, only a single bit is needed. This kind of decomposition model, which makes that there are several bound blocks on a one logic level, is called multiple decomposition (Curtis, 1962).

The essence of decomposition by Ashrenhurt – Curtis is disjoint of a bound and a free sets $Xb \cap Xf = \varnothing$. Decomposition process may undergo optimization by using non-disjoint decomposition (Dubrova, 2004; Dubrova et al., 2004). The essence of non-disjoint decomposition is attaching a part of variables to a bound set as well to a free set that create a third variable coshared set $Xb \cap Xf = Xs$. It turns out that the variables from the set *Xs* may be switching functions and replace a part of bound functions g. The essence of searching for non-disjoint decomposition for SMTBDD was presented in the work (Kubica and Kania, 2017b) In the analyzed case from figure 2, searching for non-disjoint decomposition will be proceeded in the same way as shown in figure 3.



x2x3x4	000	001	011	010	110	111	101	100
c	t	t	t	u	u	t	t	t
d	t	t	u	t	t	u	u	u
	A	A	B	C	C	B	B	B

$\upsilon(X_f|X_b)=3$

x2x4	00	01
c	t	t
d	t	t
	A	A

	11	10
c	t	t
d	u	u
	B	B

$\upsilon(X_f|X_b)=2$

x2x4	01	00	10	11
c	t	u	u	t
d	u	t	t	u
	B	C	C	B

$\upsilon(X_f|X_b)=2$

Fig. 3. The essence of searching for non-disjoint decomposition in SMTBDD: a) the partition of a root table, b) a gained partition

A starting point of searching for non-disjoint decomposition is a classic disjoint decomposition by Ashrenhurst – Curtis. The question is, is there a possibility to replace one of two bound functions by an input variable (Figure 2). Figure 3a illustrates the situation whether the variable *x3* may fulfill the role of a switching function. This variable causes the partition of a root table into two tables for *x3=0* and for *x3=1*. Column complexities *υ(Xb|Xf)*, obtained after tables' partition, give 2 in both cases and are lower from the complexity of an original table *υ(Xb|Xf) = 3*. It means that the variable *x3* may replace one of two bound functions. Searching for non-disjoint decomposition gives the logic structure presented on figure 3b. As the number of bound function is lower than on figure 2d, in order to carry out a bound set, a lower number of logic blocks will be needed.

Assigning variables to separate sets in the case of BDD, depends on variable ordering in BDD. In order to find the solution that would give the best technology mapping to blocks that have configurable abilities, multiple change of variable ordering in BDD is needed. In the case of decomposition in which there are many cutting lines, it is key to determine which variables are ascribed to particular extracts. Variable ordering inside a single extract does not matter.

Thus, it is essential to choose such a decomposition path to match it to technology limitations of logic blocks included in particular families of FPGA circuits.

3. TECHNOLOGY MAPPING OF FUNCTIONS IN MODERN LOGIC BLOCKS OF FPGA CIRCUITS

The essence of an efficient technology mapping in configurable logic blocks is the choice of the cutting line in BDD so as the number of the obtained bound sets is equal to the number of inputs (*n*) of configurable logic blocks. This approach limits the number of inputs of a logic blocks that were not used. It leads to a better usage of a given block. Configurable logic blocks enable to slightly change the number of inputs what makes it easier to choose cutting levels. From the point of view of decomposition carried out using BDD, to main parameters will decide about the quality of technology mapping. The first concerns the number of separate sets connected with cutting levels of a diagram. The second one is the number of needed bound functions necessary to replace a given extract of BDD where it is aspired to limit their number.

The most basic technology mapping is the mapping in blocks of LUT type. In the case of these blocks, the ability to change the number of inputs is strongly limited. Let us denote the number of inputs of a LUT block as *k*. Technology mapping methods may be proposed for LUT blocks for decomposition carried out by using a single as well several cutting lines.

In the case of decomposition carried out by using a single cutting line, the cardinality of a bound set *card(Xb)* corresponds to the number of variables associated with the nodes above the cutting line. The level of cutting, counting from the root, is chosen in the way *card(Xb)* is equal to the number of inputs *k* in a LUT block. For such a cutting line, it is searched for a variable ordering in BDD so as to gain the possible lowest number of cut nodes and what is more, the number of bound functions. In the case of decomposition carried out using several parallel cutting lines, mapping to LUT blocks that have *k* inputs will be carried out by an appropriate choice of 'distance' between neighboring cutting lines in a BDD diagram. The extract of a diagram, which is placed between neighboring cutting lines, should be associated with *k* variables. After choosing appropriate cutting lines, it is necessary to search for such variables ordering so as column complexity of gained root tables will be as small as possible. Similarly, as in the previous cases, it is aspired to limit the number of bound functions. For decomposition carried out by both using a single cutting line as well as several cutting lines, effectiveness of mapping may be gained by searching for non-disjoint decomposition. Moreover, it results with reduction of needed LUT blocks.

Configurable logic blocks include several LUT blocks inside. It makes them more flexible taking the number of inputs into account. Depending on the manufacturer, logic blocks may be configured in many ways. FPGA circuits by Intel (Intel, 2016) such as Stratix, Cyclone, etc. include complex logic blocks ALM (Adaptive Logic Module) in their structures. These blocks may be configured in the way they have up to 8 inputs (LUT8/1). LUT blocks, included in ALM, may also work independently, leading to the configuration 2 x 4 inputs (LUT4/2). Generally, there is a series of indirect configurations in which a part of inputs of an ALM block is co-shared between separate LUT blocks. The essence of complex configurations of ALM blocks, is enabling these blocks to operate in particular modes such as arithmetic or dedicated to the usage of DSP. In the case of FPGAs developed by Xilinx (Xilinx, 2016), the situation is similar. Configurable logic blocks CLB (Configurable Logic Block) in FPGA, series 7 are based on 6-input LUT blocks that may be configured as blocks that have 5 inputs and 2 independent outputs.

Taking into account substantially better possibility of configuration of the number of inputs of complex logic block and that they may have several independent outputs, the question is how to lead decomposition to ensure the best mapping of a function in complex logic blocks. Let us consider the situation in which one of possible configurations of a complex logic block, there are two independent LUT blocks that have *k* inputs and *n-k* inputs, respectively. It is shown in figure 4.

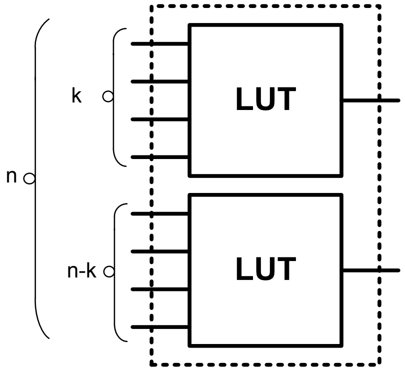


Fig. 4. Configuration of a logic block as two independent LUT blocks

The number of inputs of separate LUT blocks, belonging to a complex logic block, automatically imposes cutting lines as presented on figure 5.

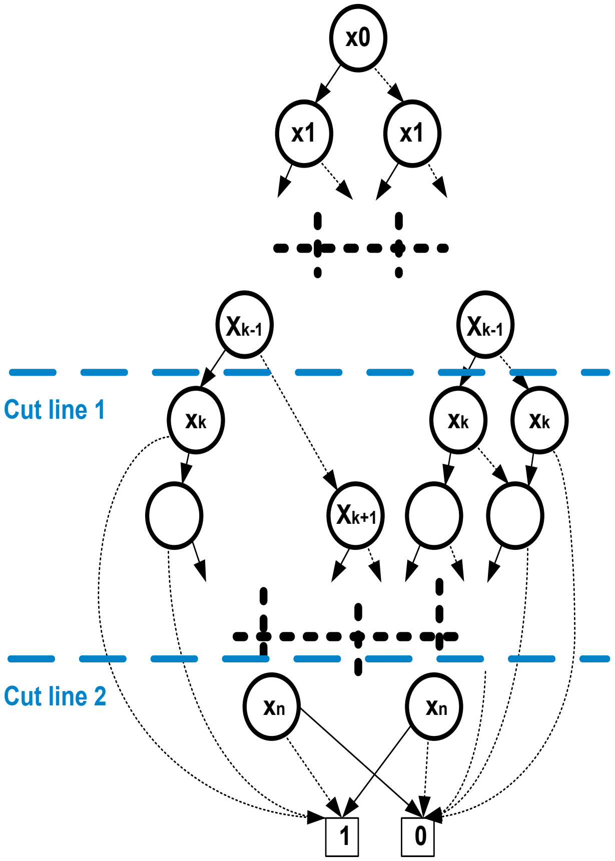


Fig. 5. Cutting of a BDD diagram for a logic block configured as shown in figure 4

The extract placed above the cutting line 1 is associated with a bound set *Xb1*. The extract placed between cutting lines 1 and 2 is connected with a bound set *Xb2*. The obtained partition model will lead to multiple decomposition. The number of a bound at *Xb1* is *card(Xb1) = k* and corresponds with the number of inputs of one of LUT blocks from figure 4. The number of a bound set *Xb2* is *card(Xb2) = n – k* and corresponds with the number of the second of LUT blocks from figure 4.

Such a choice of a cutting line leads to the situation in which both bound blocks will be mapped in the same configurable logic block. Obviously, the number of blocks needed to carry out a bound set depends on the number of bound functions. Besides, imagine the opposite situation in which cutting lines were chosen on the levels ensuring *card(Xb1) = n – k* and *card(Xb2) = k*. It will also lead to mapping in configurable logic blocks from figure 4. Thus, the number of bound functions decides which choice of cutting lines is more effective.

Multiple decomposition, performing appropriately, carried out using multiple cutting method and completed with non-disjoint decomposition, may give efficient mappings in complex configurable logic blocks. A coherent synthesis strategy may be proposed after having completed it with the techniques of combining functions (Kubica et al., 2017) and the methods of assessing the efficiency of mapping (Kubica and Kania, 2017a). This strategy was called MultiDec and was described in the papers by (Kubica and Kania, 2017a; 2017).

It is essential now to compare the obtained results with the results for commonly used commercial tools.

4. EXPERIMENTAL RESULTS

Developed decomposition strategy was applied to the selected set of benchmark circuits (Collaborative Benchmarking Laboratory). The benchmark set is delivered in PLA format. The PLA format was converted to Verilog HDL using continuous assignment statements. For each benchmark were generated two descriptions. The first one directly corresponds to PLA content. It uses continuous assignment statement of Verilog HDL.

In the table 1 they were marked with the X header. The second form was obtained applying MultiDec strategy. The circuit was partitioned into multiple level structure (when needed). Each assigned statement corresponds to the content of a LUT. In order to preserve synthesized structure, internal nodes are forced to be kept by synthesis tools with respective directives (Xilinx, 2013). Synthesis results obtained with use of MultiDec strategy are gathered in columns with header MD.

Prepared two forms of benchmark circuits descriptions was synthesized with commonly used tools. For comparison purposes there were chosen: Xilinx ISE 14.7, Xilinx Vivado 2016.4, Quartus and Synplify 2014. The ISE package is mature development system but it is still widely used. The Vivado design environment is dedicated for the newest FPGA families belonging to 7th series devices and Zynq (SoC line devices). The target device is Artix XC7A100 that is low power version of series 7 FPGAs.

The Quartus integrated design environment is delivered by Intel (formerly Altera) for manufactured lines of FPGAs. The synthesis was dedicated for Cyclone 10 GX family. For synthesis experiments was chosen 10CX220YF chip. The Synplify tool is developed by Synopsys that is not manufacturer of FPGA devices.

The utilized resources are reported by number of LUTs (ALUT) or Adaptive Logic Modules (ALM – Cyclone 10 GX) that are gathered in the table 1. There was gathered results of synthesis of MultiDec strategy compared with results of mentioned development packages.

In the initial columns were described benchmarks, its name, number of inputs and outputs. In the following columns are gathered synthesis results. Additionally number of circuit levels are reported for MultiDec, Quartus and Synplify.

In the last row of table 1 are gathered total count of LUTs and number of circuit levels for respective systems. The comparison of total number of LUTs is shown in figure 6 while comparison of total number of logic levels is shown in figure 7.

Additionally, there was shown total number of LUTs (Fig.6) and logic levels (Fig.7) obtained in synthesis process supported with MultiDec circuit refactoring.

Table 1. Direct comparison of different logic systems

“MultiDec” – the results predicted by MultiDec, “X” - synthesis results without MultiDec, “MD” – synthesis results with MultiDec, “In” – the number of inputs, “Out” – the number of outputs, “LUT” – the number of LUT blocks, “lev.” – the number of logic levels.

					ISE		Vivado		Quartus				Synplify			
			MultiDec		X	MD	X	MD	X		MD		X		MD	
Name	In	Out	LUT	Lev.	LUT	LUT	LUT	LUT	LUT	Lev.	LUT	Lev.	LUT	Lev.	LUT	Lev.
5xp1	7	10	8	2	14	13	12	11	16	2	13	2	10	2	8	2
b12	15	9	12	2	16	20	14	19	15	2	25	3	12	2	14	2
cm85a	11	3	5	2	6	8	6	8	10	2	10	3	7	3	6	2
con1	7	2	1.5	1	2	2	2	2	2	1	2	1	2	1	2	1
f51m	8	8	5.5	2	13	10	10	8	19	3	10	2	8	2	6	2
inc	7	9	9	2	11	13	10	12	13	2	14	3	10	2	10	2
ldd	9	18	19	2	26	27	19	22	29	4	30	2	14	3	21	2
misex1	8	7	6.5	2	9	9	8	8	7	1	9	2	8	2	7	2
pcle	19	9	9	3	12	12	9	12	11	2	11	3	9	3	10	3
rd73	7	3	3	2	19	6	6	6	9	2	6	2	6	3	6	2
rd84	8	4	6	2	13	9	11	8	69	4	9	3	10	2	6	2
sqn	7	3	5	2	6	9	6	9	9	2	9	2	6	1	6	2
t481	16	1	4.5	4	4	8	7	8	6	3	9	4	4	3	6	4
x2	10	7	6	2	11	11	8	9	13	2	17	3	9	3	7	2
Sum:			99	30	162	157	128	142	228	32	174	35	115	32	115	30

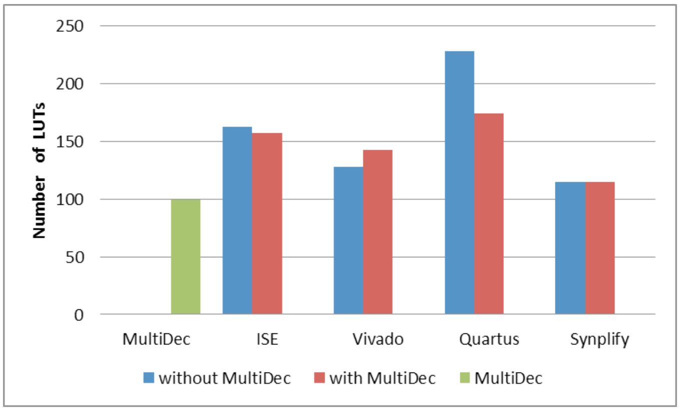


Fig. 6. The comparison of the logic systems in respect of number of LUTs

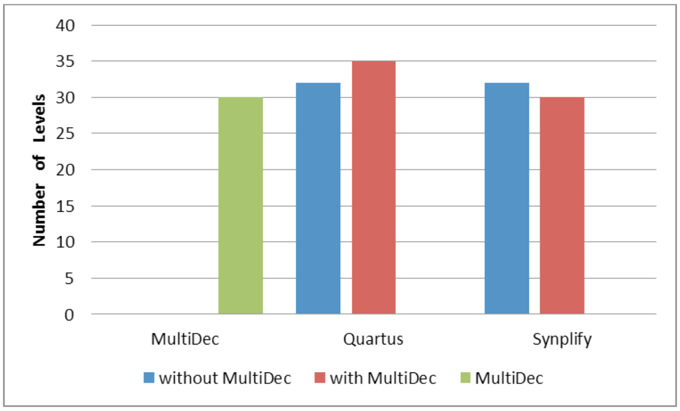


Fig. 7. The comparison of the logic systems in respect of number of logic levels

Comparison of synthesis result reveals reduction of logic resources when circuit structure has been refactored using MultiDec startegy. The significant reduction is observed in Quartus and ISE toolsets. The synthesis process supported with MultiDec strategy reduces overall logic resources requirements about 23% for Quartus. The Synplify synthesis tool delivers as good as results independently from applying or not MultiDec refactoring. The Vivado synthesis delivers overall slightly better results without MultiDec circuit partitioning.

Implementation with MultiDec results with slightly increase (10%) of the number of logic levels for Quartus. Reduction of logic levels with MultiDec strategy is observed in case of the Synplify tool.

Comparison of synthesis results for particular systems supported with MultiDec procedure proves possibility of reducing the total number of logic blocks and reduction of logic levels than in case of direct PLA benchmark implementation. There is observed a strong dependency between synthesis results and Verilog HDL description refactored with MultiDec and synthesis constraints. The logic equation based representation results with worse implementation results than theoretically expected in the MultiDec tool (see Tab.1). There is need to improve the method of representing refactored circuit in Verilog HDL to protect its structure against undesired modifications.

5. CONCLUSIONS

One of the problems of logic synthesis and mapping for FPGA devices is an efficient partitioning of combinatorial circuit to available logic blocks. This problem becomes more complex and important when number of possible configurations variants of a block is taken into account. The paper brings and idea of functions set mapping into LUT based FPGAs It was implemented in MultiDec system that enables mapping to desired architecture of LUT blocks including its alternative operations.

The experience gained during design of digital circuits on programmable logic devices with use of different tools show an evidence of substantial synthesis result improvement. Synthesis results are strongly dependent of circuit description method that vary among used tools.

The MultiDec system implements described algorithms and enables instant assessment and continues improvement particular algorithms. It results with many improvements in synthesis algorithms and improvements of technology mapping. Carried out experiments are the clear evidence that there exist a possibility of improving synthesis results of commercially available tools especially by reducing a circuit area. Nevertheless, there is observed continues progress of synthesis result quality in commercially available tools especially their efficiency of programmable resources fitting.

REFERENCES

Ashenhurst, R.L. (1957). The Decomposition od switching functions, *Proceedings of an International Symposium on the Theory of Switching*, 1957

Abouzeid, P., Babba, B., Crastes, M., Saucier, G. (1993). Input-Driven Partitioning Methods and Application to Synthesis on Table-Lookup-based FPGAs, *IEEE Trans on CAD*, 12, No. 7, pp. 913-925

Babu, H. Sasao. T., (1998). Shared multi-terminal binary decision diagrams for multiple-output functions, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, **Vol.81, No.12**, pp. 2545-2553

Berkeley Logic Synthesis Group: *ABC: A System for Sequential Synthesis and Verification*, Dec. 2005 Available: http://www.eecs.berkeley.edu/~alanmi/abc

Bryant, R.E., (1986). Graph Based Algorithms for Boolean Function Manipulation, *IEEE Transactions on Computers* vol.C-35, no. 8/1986, pp. 677 - 691

Collaborative Benchmarking Laboratory, Department of Computer Science at North Carolina State University, http://www.cbl.ncsu/edu/

Curtis, H.A., (1962). *A New Approach to the Design of Cwitching Circuits*, D. van Nostrand Company Inc, New York

Dubrova, E. (2004). A Polynominal Time Algorithm for Non-Disjoint Decomposition of Multi-Valued Functions, *34th International Symposium on Multiple-Valued Logic*, pp. 309-314

Dubrova, E., Teslenko, M., Martinelli, A., (2004). On Relation Between Non-Disjoint Decomposition and Multiple-Vertex Dominators, *Circuits and Systems*, 2004. ISCAS '04, **Vol.4**, pp. 493-496, 2004

Fiser, P., Schmidt, J., (2012). On Using Permutation of Variables to Improve the Iterative Power of Resynthesis, *in Proc. of 10th Int. Workshop on Boolean Problems (IWSBP),* Freiberg (Germany), pp.107-114

Fiser, P., Schmidt, J., (2009). The Case for a Balanced Decomposition Process, *in Proc. of 12th Euromicro Conference on Digital Systems Design (DSD),* Patras (Greece), pp. 601-604

Francis, R.J., Rose, J., Chung, K., (1990). Chortle: A Technology Mapping Program for Lookup Table-Based Field Programmable Gate Arrays, *27th ACM/IEEE Design Automation Conference*, pp. 613-619

Intel, (2016). Stratix V Device Handbook, Volume 1: Device Interfaces and Integration

Jozwiak, L., Chojnacki A., (2003). Effective and Efficient FPGA Synthesis through General Functional Decomposition, *Journal of Systems Architecture*, **Vol. 49, No. 4-6**, pp. 247-265

Kubica M., Kania M., (2015). SMTBDD: New Concept of Graph for Function Decomposition, *IFAC Conference on Programmable Devices And Embedded Systems* PDeS, Krakow, **Vol. VII, No. 519**, pp. 61-66

Kubica, M., Kania, D., (2017a). Area-oriented technology mapping for LUT-based logic blocks, *International Journal of Applied Mathematics and Computer Science,* **Vol. 27, No. 1**, pp. 207-222

Kubica, M., Kania, D., (2017b). Decomposition of multi-output functions oriented to configurability of logic blocks, *Bulletin of the Polish Academy of Sciences – Technical Sciences*, **Vol. 65, No. 3,** pp. 317-331

Kubica M., Opara A., Kania D., (2017). Logic synthesis for FPGAs based on cutting of BDD, *Microprocessor and Microsystems,* **Vol. 52**, pp. 173-187

Minato, S., (1996). *Binary Decision Diagrams and Applications for VLSI CAD*, Kluwer Academic Publishers

Murgai, R., Shenoy, N., Brayton, R.K., Sangiovanni-Vincentelli, A. (1991). Improved Logic Synthesis Algorithms for Table Look up Architectures, *ICCAD-91*, Santa Clara, CA, pp. 564-567

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

Vemuri, N., Kalla, P., Tessier, R., (2002). BDD-based logic synthesis for LUT-based FPGAs, *ACM Trans. Design Autom. Electron. Syst.,* **Vol. 7, No. 4**, pp. 501-525

Xilinx, (2013) UG687, XST User Guide for Virtex-6, Spartan-6, and 7 Series Devices

Xilinx, (2016). 7 Series FPGAs Configurable Logic Block - User Guide (UG414)