FPGA LUTs for a Logic Systems

Tyurin Sergey, Prokhorov Andrey, Vikhorev Ruslan

***Abstract*—** FPGA logic based on Look up Tables (LUTs). However, LUT calculates only one logic function in the perfect disjunctive canonical forms (PDCF) for this configuration. The paper proposed the concept of the logic advanced LUTs in three main areas. The first area is Double LUT (DLUT), computes two functions simultaneously with inactive transmission transistors subtree. The second area is DC LUT through inverse this tree to implement the decoder DC, which allows computes whole system of the logic functions. Such technique can significantly reduce hardware expenses for logic systems. The third area is DNF-LUT, which allows the calculations of the system functions in disjunctive normal form (DNF) and else more significantly reduces large number of variables LUTs hardware costs. We explored the models of the proposed concepts in the NI Multisim 10 by National Instruments Electronics Workbench Group. The paper analyzes the assessments of the complexity of the LUT, the conclusions about the effectiveness of the proposed solutions.

***Keywords*—** look up table; FPGA; perfect disjunctive canonical forms.

I. INTRODUCTION

FPGA chips are widely used in computer technology [1-3]. There are quite a large number of energy-saving methods of configuring FPGA [1, 2] for example, such as energy-efficient mapping and clocking, unused blocks power down and others. In Ph.D. Dissertation [4] suggested an FPGA post-fabrication component-specific mapping and an optimized architecture taking into account the characteristics of individual transistors, identified during the operational phase. It uses minimum energy/operation indicator. However, the expansion of the actual FPGA logic capacity for energy efficiency presented in the available sources not full. For this, it is possible to use logic optimization [5-7]. This particularly applies to the implementation of logical systems, an example of which given by CPLD [8].

A goal of the paper is research and development logic elements-LUTs by reducing the complexity of the realization of logic functions of a large number of arguments. It presented LUT PDCF technique – DLUT & DC LUT, computes two and more functions in perfect disjunctive canonical forms (PDCF) simultaneously. Secondly is devoted LUT DNF technique by analogy with Programmable Logic Array (PLA). Described the comparison of the number of transistors to implement system of the logic functions in the known LUT and in all proposed LUT.

II. METHODOLOGY

A. LUT FPGA Calculates Two Functions Simultaneously

Consider the LUT model on the two variables (2-LUT), configured to calculate the exclusive OR function is shown in Fig. 1.

S. F. Tyurin, Perm National Research Polytechnic University, Perm, Russia (e-mail: tyurinsergfeo@yandex.ru).
A. S. Prokhorov, Perm National Research Polytechnic University, Perm, Russia (e-mail: npoxop007@yandex.ru).
R. V. Vikhorev, Perm National Research Polytechnic University, Perm, Russia Japan (e-mail vihrusvla@mail.ru).

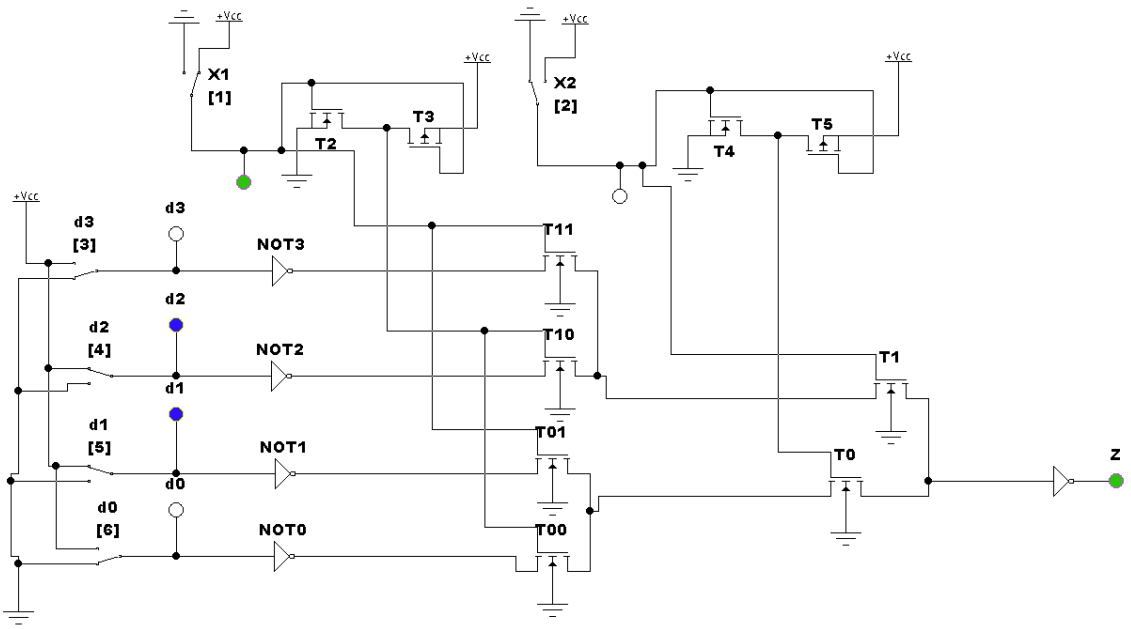

Figure 1. The 2-LUT model configured to calculate the exclusive OR function.

LUT is allocated only one logic function *z*, customizable by the user by downloading the configuration memory SRAM (d0-d3). At the same time during the computation is always activated only half the tree transistors (T0&T00 or T01 when $x_2 = 0$; T1&T10 or T11 when $x_2 = 1$). This creates the conditions for the use of idle half of the transistors with the introduction of another pair of leading variable. However, this requires connectivity configuration memory SRAM, which stores the settings of the second function. The corresponding truth table is stored "backwards" compared to the truth table of the first function, Fig. 2.

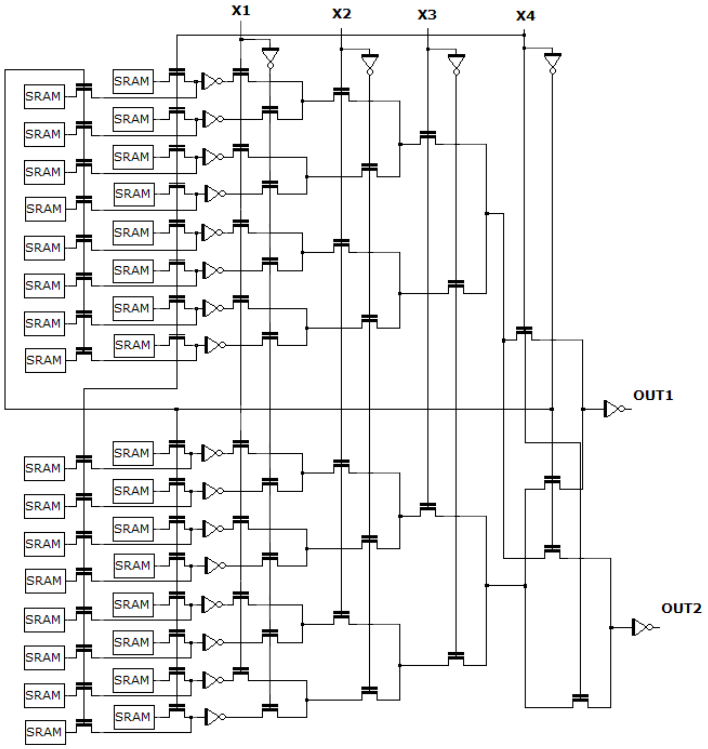

Figure 2. Duble4-LUT (D4-LUT) calculates two functions simultaneously

The simulation of the Duble2-LUT (D2-LUT) calculates two function $z_1 = x_1 \leftrightarrow x_2$, $z_2 = x_1 \oplus x_2$, $x_1 = x_2 = 0$ in the system NI Multisim 10 is presented in Fig. 3-5. To use the second half of the tree transistors introduced additional transistors leading variable T0.1 and T1.1 and transistors connection settings first function, T00.1, T01.1, T10.1, T11.1, second function T00.2, T01.2, T10.2, T11.2. The setting will simulate constant connection - supply pins V_{cc} and Ground -"zero volts". Simulation confirms the efficiency of the proposed scheme DLUT.

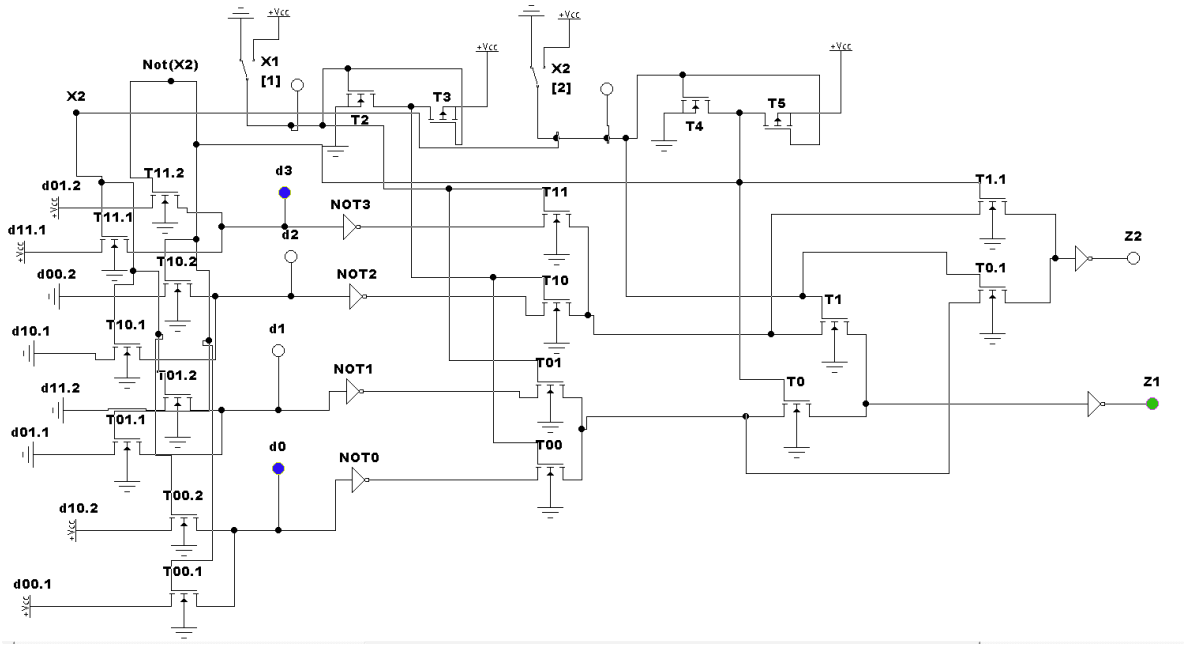

Figure 3. The simulation of the D2-LUT

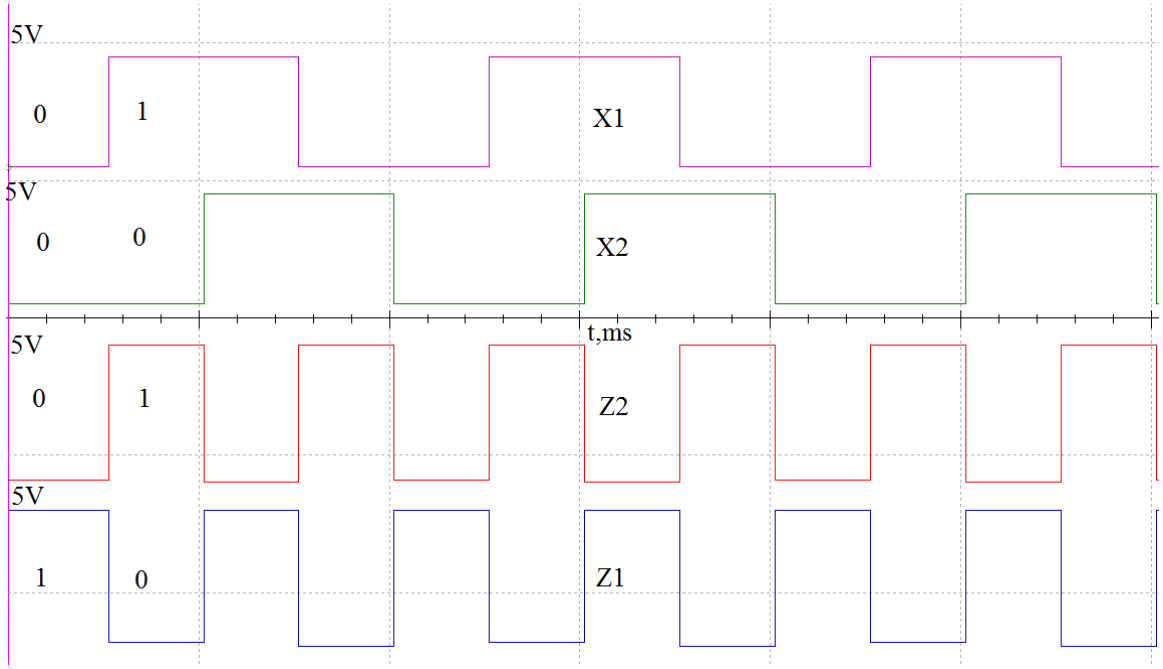

Figure 4. D2-LUT – waveform

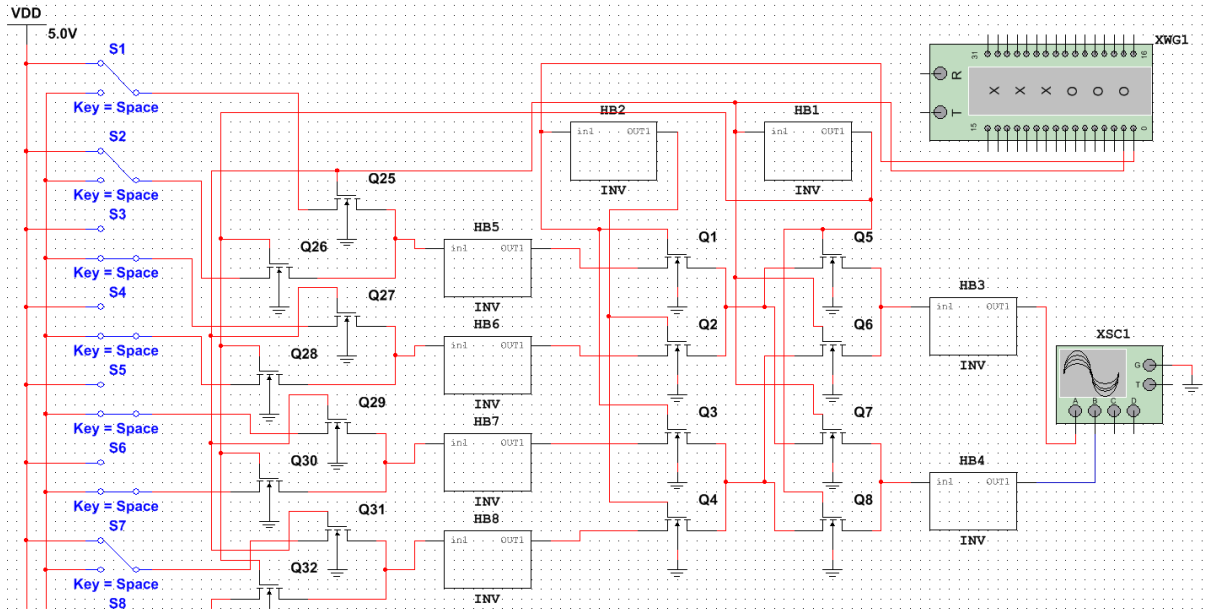

Figure 5. D2-LUT - dynamic model

B. D C LUT FPGA, Calculates System of the Logic Function

The transistor tree "reverse" structure LUT (see. Fig. 1) obtained by “reflection” on the LUT horizontally [9], Fig. 6.

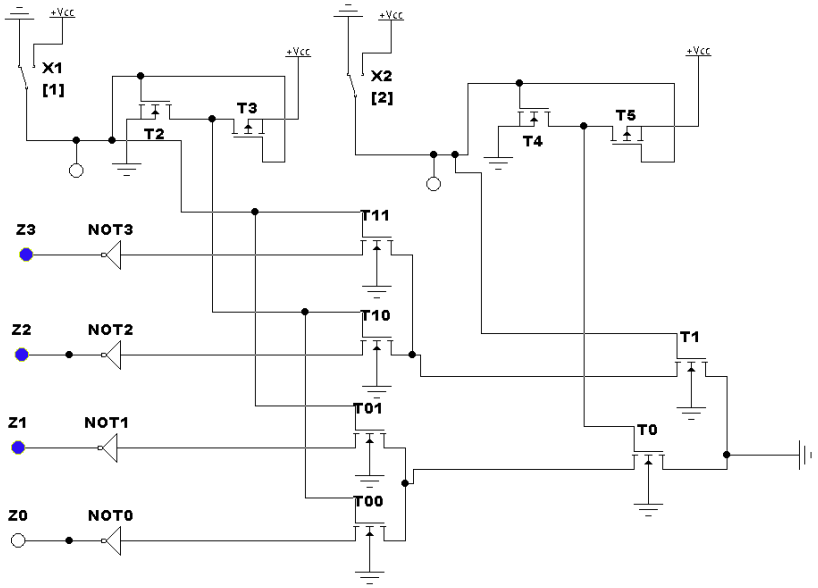

Figure 6. "Reverse" 2-LUT structure

In compliance with the design rules circuits of the transmission transistors required for a drain of each transistor T00, T01, T10, T11 (Fig. 14) to create an alternative chain, transforming its output is guaranteed, for example, in a logic “1”. The best option is to create an alternative transistor for each transmission transistor - as it presented in Fig. 7.

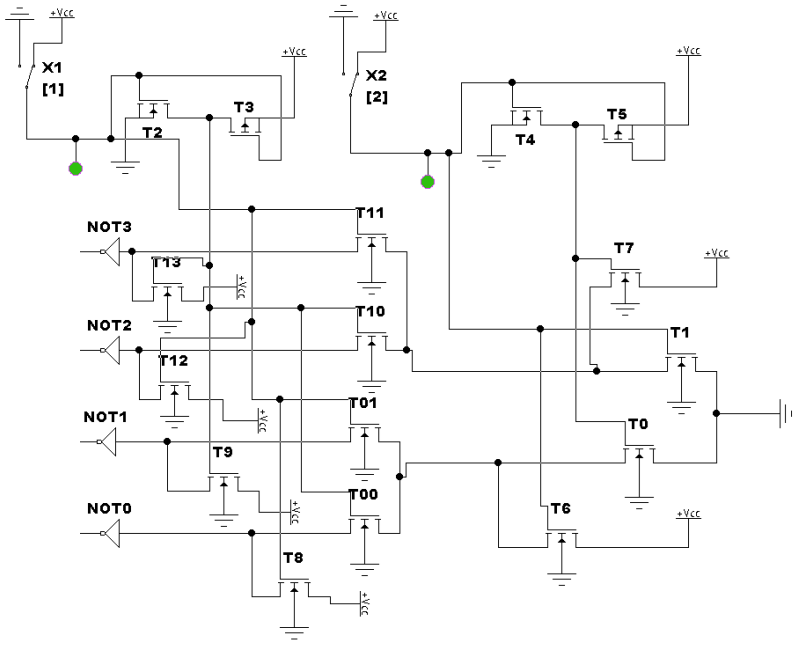

Figure 7. Reverse tree 2-LUT (DC 2-LUT) with alternate transistors T6, 7, 8, 9, 12, 13.

Get the decoder with the output function *z* (without alternative chains). Further, *m* times by combining the OR, the corresponding outputs of the outputs we get the implementation of the system of *m* n-bit logic functions based on perfect disjunctive normal forms (PDNF). DC 2-LUT with one programmable unit disjunctions configured to implement $x_1 \oplus x_2$ show Fig. 8-10.

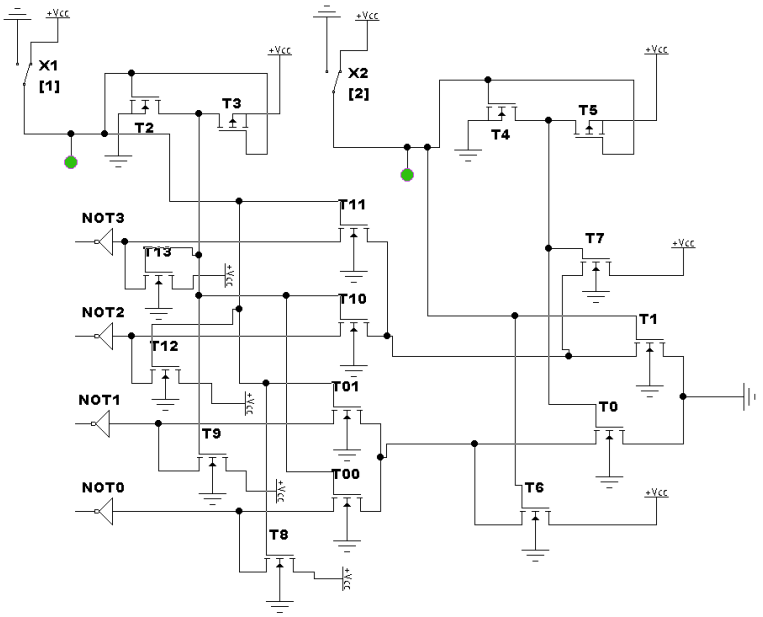

Figure 8. DC 2-LUT - static model

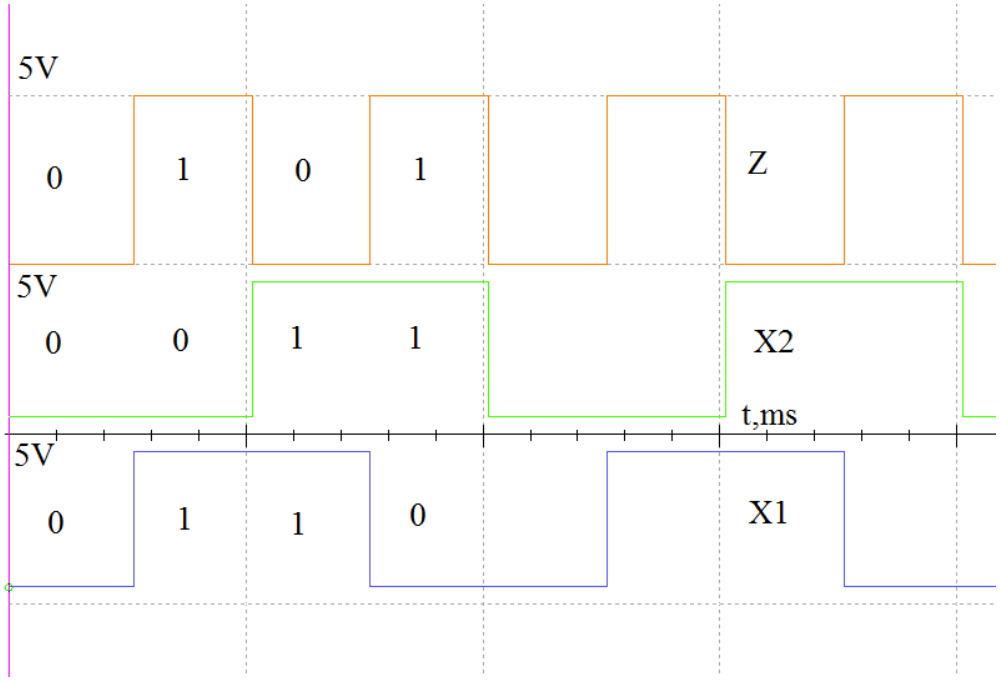

Figure 9. DC 2-LUT – waveform

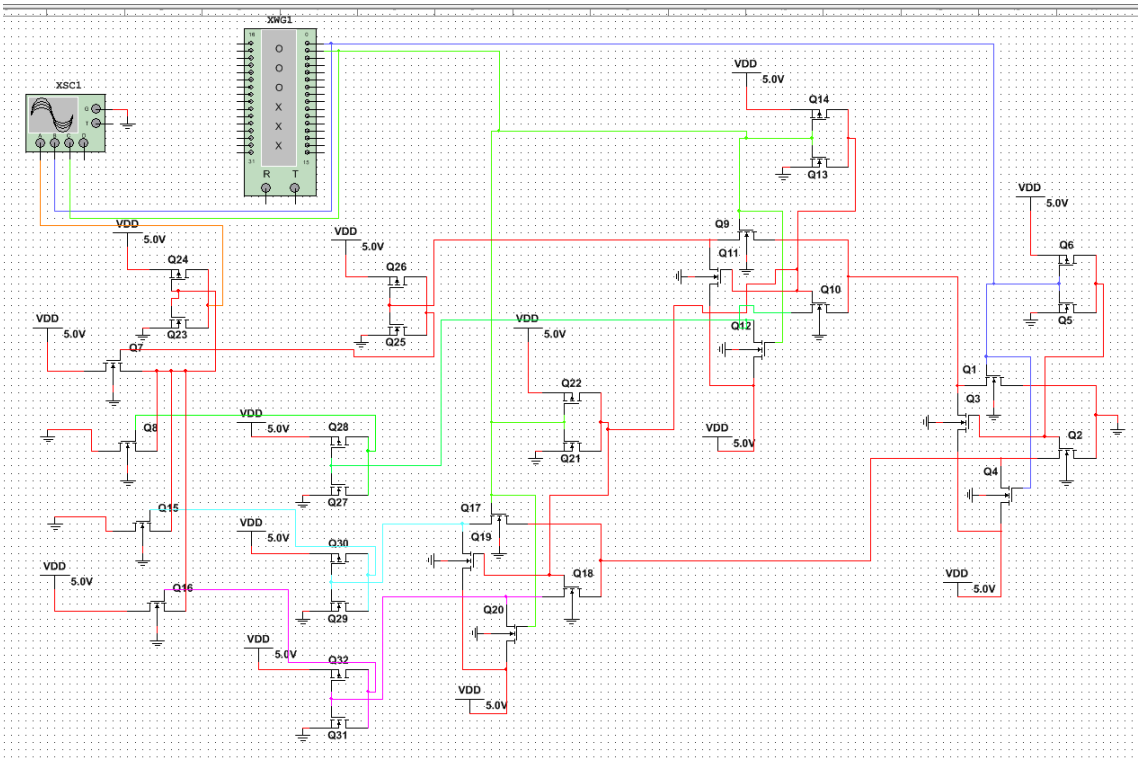

Figure 10. DC 2-LUT - dynamic model

C. Advanced LUT FPGA for Disjunctive Normal Form (DNF) of the Logic Functions Architecture DNF -LUT

The proposed new DNF-LUT [10] is a user-configurable structure similar to a programmable logic array PLA. Architecture DNF-LUT presented on the Fig. 11.

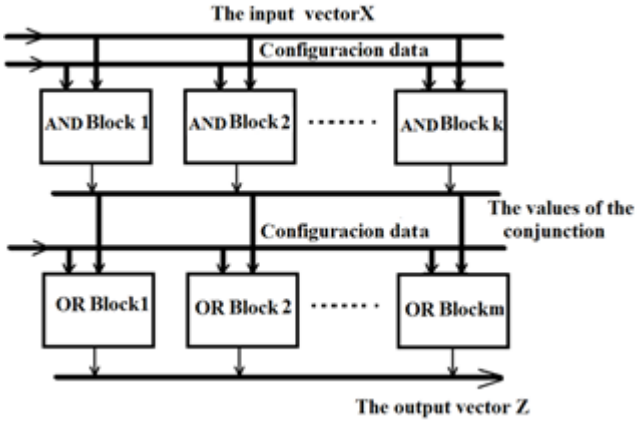

Figure 11. Architecture DNF-LUT

Thus, instead of loading the truth table, it is loading only values programmed conjunctions length *n*, where *n* - the number of variables of *m* logic functions. Occurrences *k* conjunctions in *m* functions are also programmable tuning functions. For a given input set (vector *n* variables *x*) *k* AND blocks calculate value of *k* conjunctions, which then form "an OR" value of *m* logic functions. The proposed structure of the AND DNF-LUT block is shown in Fig. 12.

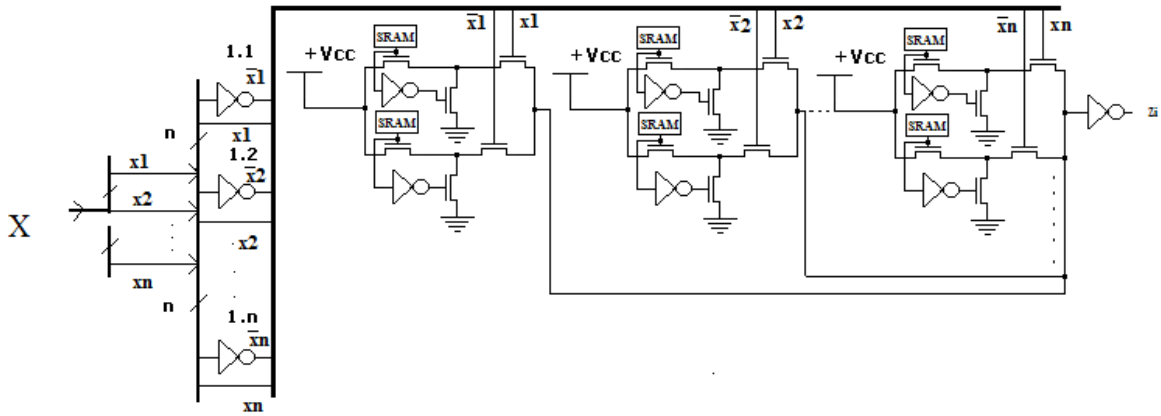

Figure 12. Architecture of the AND DNF-LUT block

One variable SRAM Setup is determined as shown in Fig. 13.

SRAM X	SRAM not X	Output 1	Output 0
1	0	When X	When not X
0	1	When not X	When X
1	1	Anyway	-
0	0	Banned	Banned

Figure 13. One variable AND DNF-LUT SRAM Setup

Thus, if the variable activated, "right", block AND, transmits a logic one signal from the input (left) to the output (right). The same occurs with the immateriality of the variable, i.e., for any value of the variable. If activated "wrong" variable, using inverters and additional transmission transistors supply to the output logical zero. If all variables are the "right" - the output z_i is a logical zero. The proposed structure of the block OR DNF-LUT show Fig. 14.

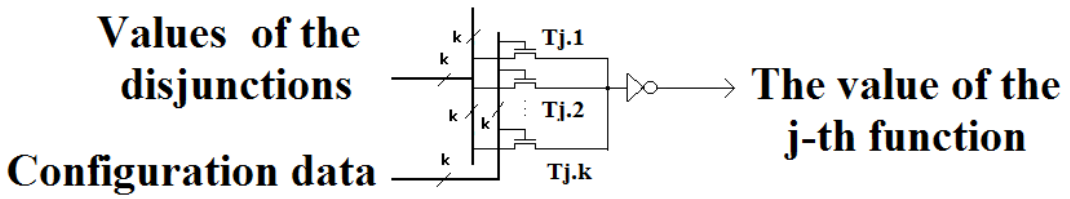

Figure 14. Architecture OR DNF-LUT block

A logical “1” at the output of the corresponding function activated when the inputs given conjunctions are zeros. The simulation of the block AND DNF-LUT executed in the system NI Multisim 10 by National Instruments Electronics Workbench Group show Fig. 15.

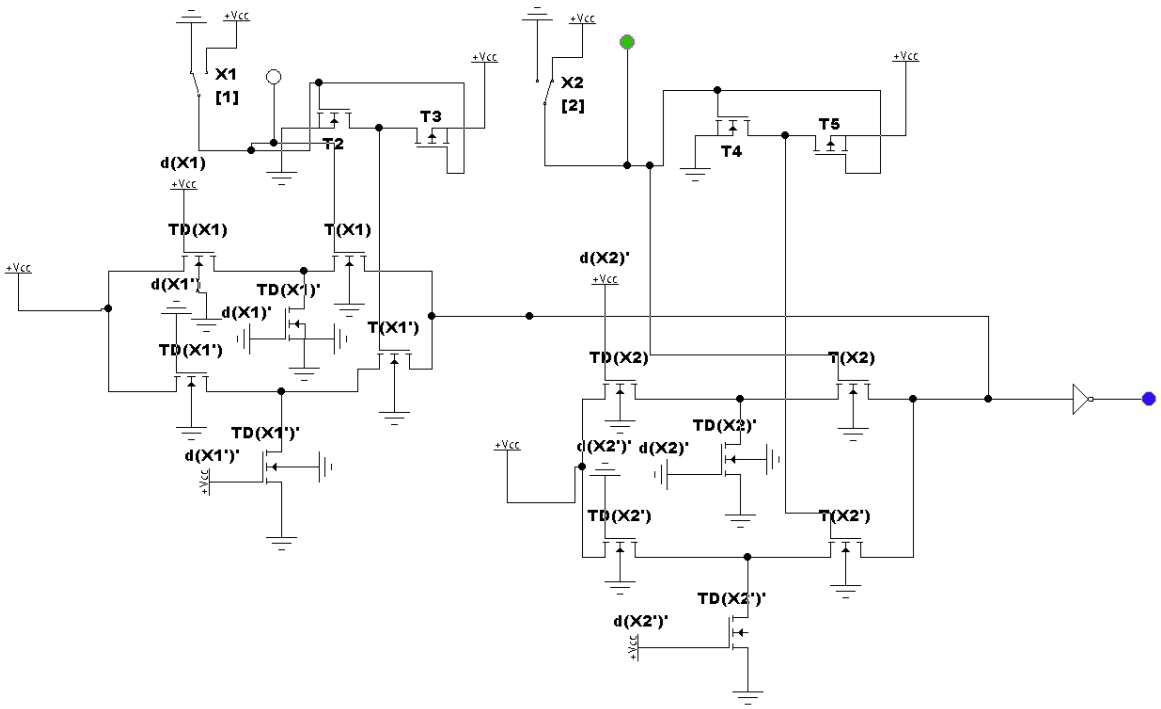

Figure 15. AND DNF-LUT Multisim model: $x_2x_1 = 0$, since $x_2 = 1, x_1 = 0$

Given a conjunction $x_2x_1 = 0$, since $x_2 = 1, x_1 = 0$. In case $x_2 = 1,\ x_1 = 1$ a conjunction $x_2x_1 = 1$ - Fig. 16.

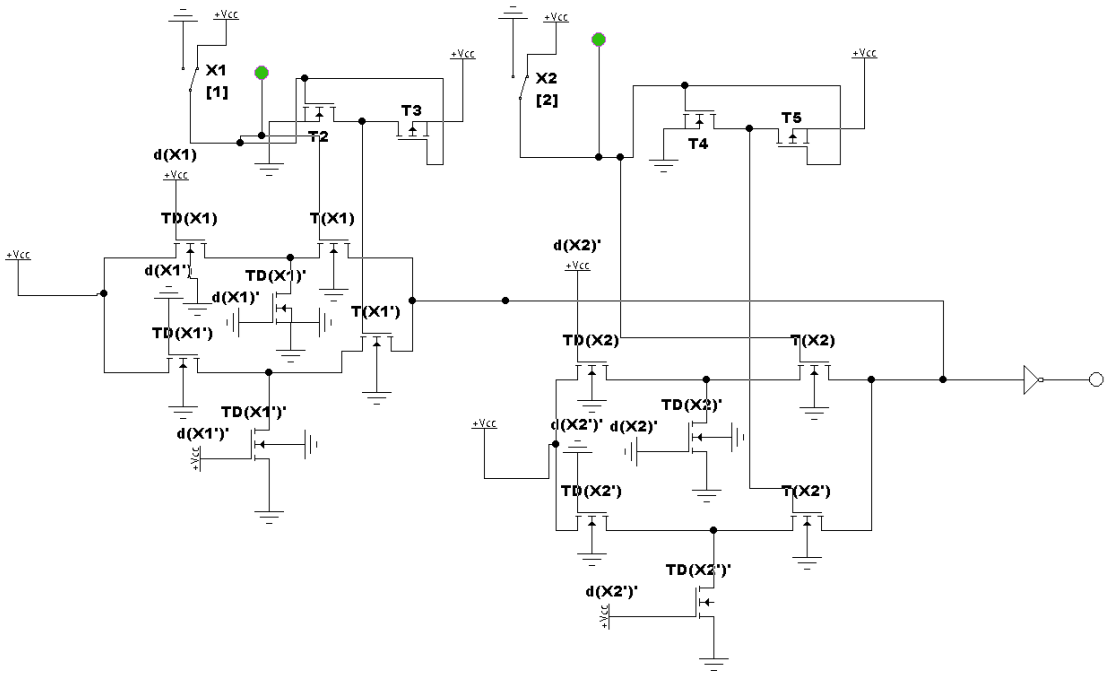

Figure 16. AND DNF-LUT Multisim model: $x_2 x_1 = 1$, since $x_2 = 1, x_1 = 1$

Checking the operation of the rest of the sets also confirms the efficiency of the proposed technical solution. The simulation of the block OR DNF-LUT in the system NI Multisim 10 by National Instruments Electronics Workbench Group presents Fig. 17-19.

Correctly, shaped value function and other combinations of k_1, k_2, k_3, k_4, modeling confirms the efficiency of the unit disjunctions.

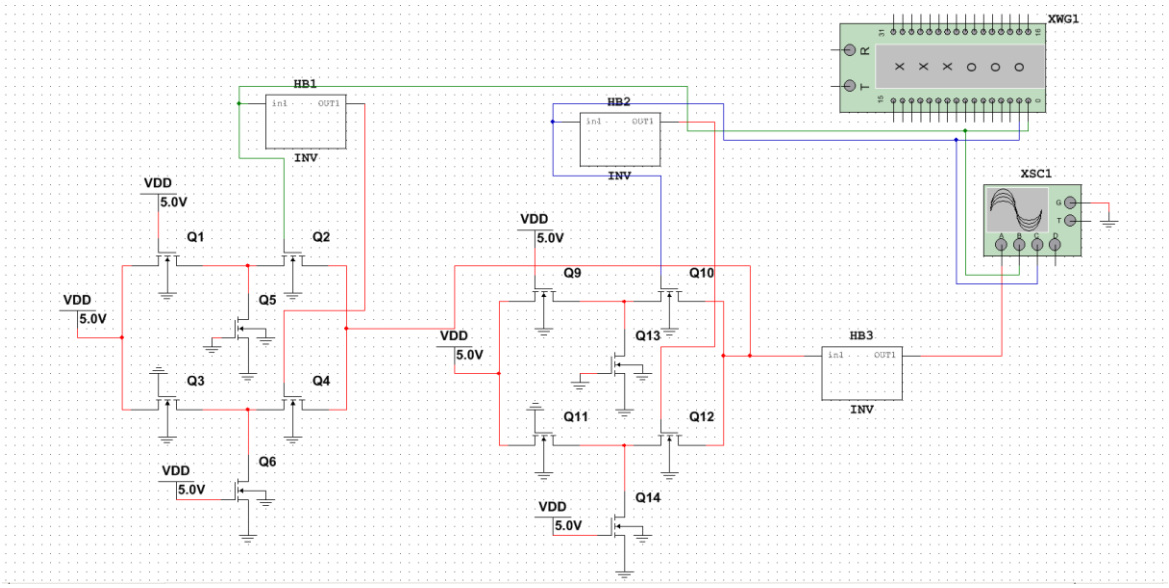

Figure 17. DNF – LUT - dynamic model

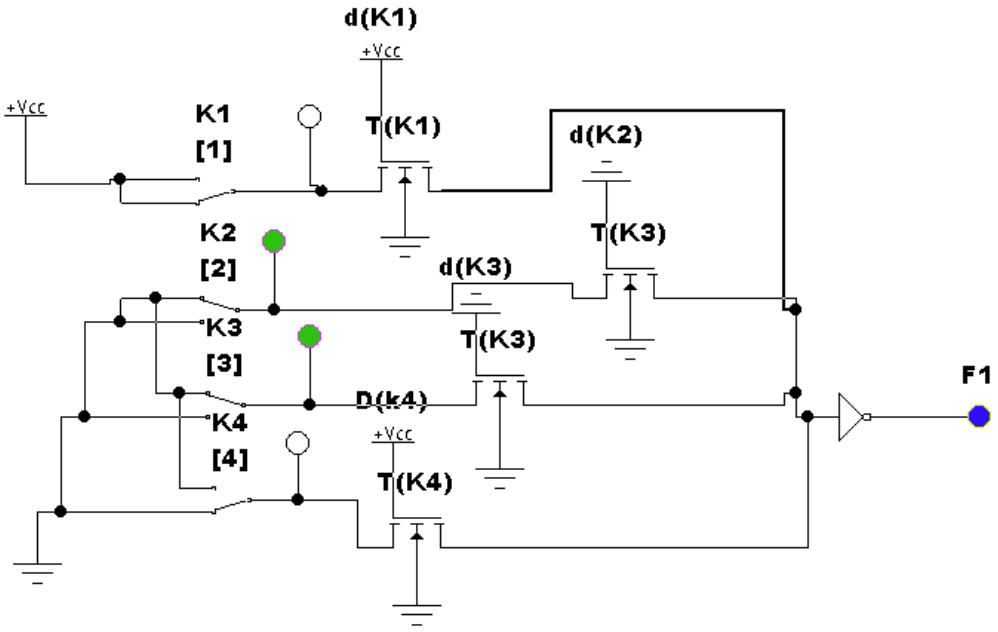

Figure 18. OR DNF-LUT Multisim model: Function $F_1 = 1$ as active k_1 and k_4.

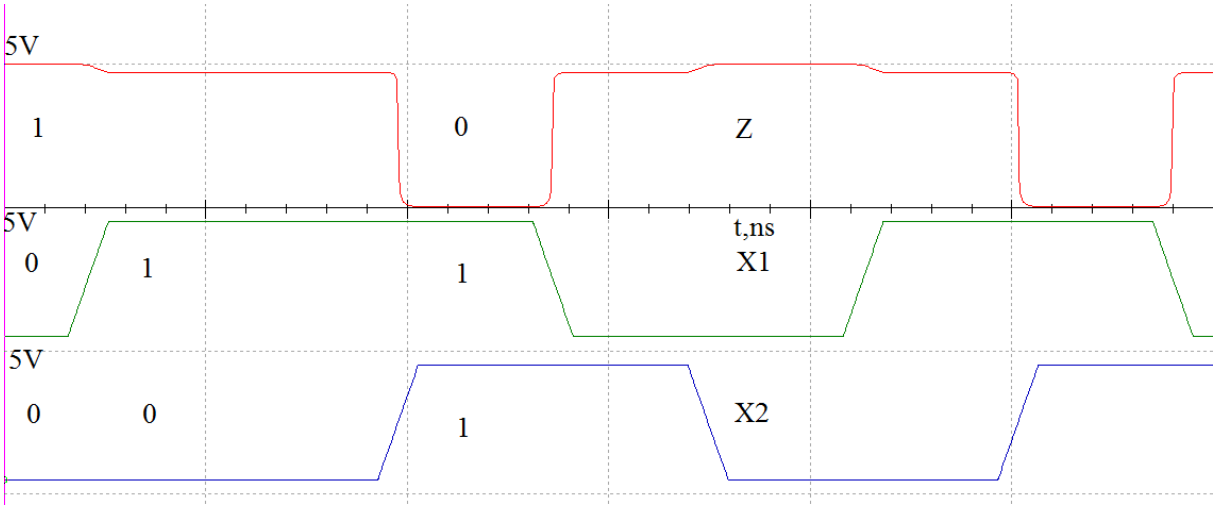

Figure 19. DNF – LUT - waveform

III. Results Of Calculations

Let k - is the dimension of the main (basic) LUT ($k \in \{1, 2, 3, 4\}$). Let us estimate the complexity of the LUT without decomposition. "Ideal" complexity can be only up to $n = 4$, not more than:

$$L_n = 2^n \cdot 8 + 2^{n+1} + 2n \quad (1)$$

With the decomposition n-tree for k LUT, we get "real" complexity:

$$L_{n.k} = 2^n \cdot 8 + (2^{k+1} + 2k) \cdot 2^{n-k} + (2^{2^{n-k}+1} + 2^{n-k+1}) + 2n \quad (2)$$

Then we get the complexity of the two logical functions:

$$2 \cdot L_{n.k} = 2[2^n \cdot 8 + (2^{k+1} + 2k) \cdot 2^{n-k} + (2^{2^{n-k}+1} + 2^{n-k+1}) + 2n] \quad (3)$$

Using Double LUT, we get:

$$L_{d-n.k} = [2^n \cdot 8 + (2^{k+1} + 2k) \cdot 2^{n-k} + (2^{2^{n-k}+1} + 2^{n-k+1}) + 2n] + 2^{n+3} + 4 \quad (4)$$

In according with the expression (1-5) the complexity of the *m* known LUT described by:

$$L_{n.k} = m[2^n \cdot 8 + (2^{k+1} + 2k) \cdot 2^{n-k} + (2^{2^{n-k}+1} + 2^{n-k+1}) + 2n] \quad (5)$$

For DC LUT we get:

$$L_{dc-n.k} = [2^n \cdot 8 + 2 \cdot \{(2^{k+1} + 2k) \cdot 2^{n-k} + (2^{2^{n-k}+1} + 2^{n-k+1})\} + 2n] + \text{ceil}(\frac{n}{4}) \cdot (2k + 4m) \quad (6)$$

Considering expressions (4) and (6), taking into account the discrete parameters we get:

$$L_{dnf} = k(22n + 2) + m(k + 2) + 6m(k + \frac{n}{4} \cdot (2k + 4m)) + 2n \quad (7)$$

Where $k(22n + 2)$ - is a complexity (in the number of the transistors) of the AND blocks; $m(k + 2)$ - is the complexity of the OR blocks; $6mk$ - is the complexity of the OR blocks setting; $2n$ - is the complexity of the inverters for variable inputs; $\frac{n}{4} \cdot (2k + 4m)$ - is the complexity of the blocks in accordance with the restriction [11].

The comparison of the number of transistors to implement system of the logic functions in the known LUT and all proposed LUT at $m = 2$ are presented Fig. 20.

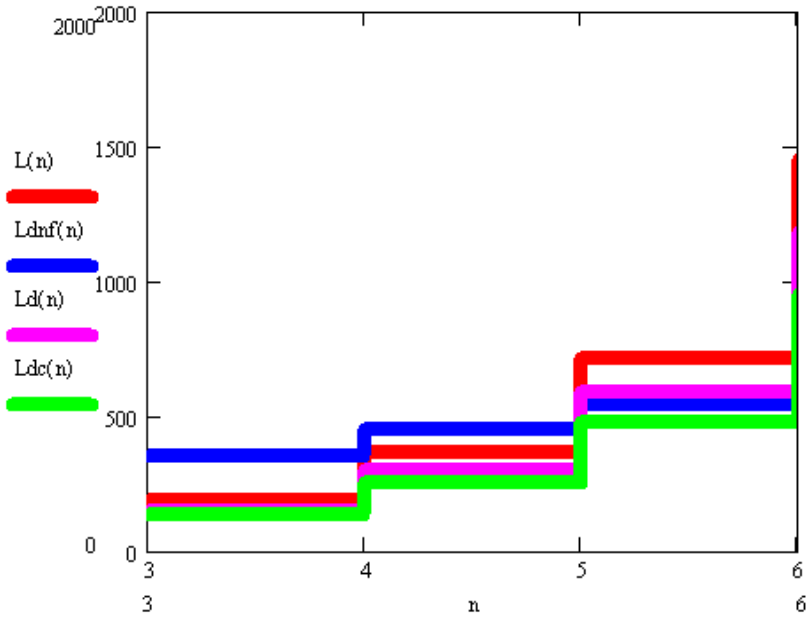

Figure 20. Comparison of the complexity of the *m* known LUT ($L(n)$) and proposed DLUT ($L_d(n)$), DCLUT ($L_{dc}(n)$), DNF-LUT ($L_{dnf}(n)$), $m = 2$

Thus a small number of functions inefficiently version DNF-LUT ($L_{dnf}(n)$). However, when calculating the 4 functions DNF-LUT ($L_{dnf}(n)$) ahead DCLUT ($L_{dc}(n)$) by 6 variables, Fig. 21.

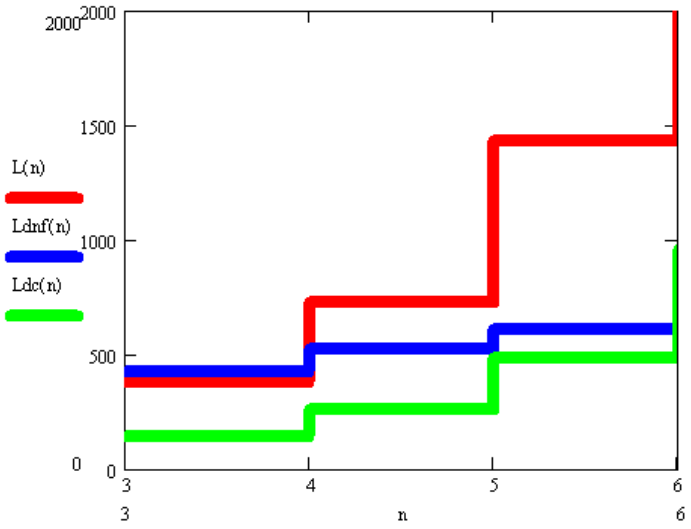

Figure 21. Comparison of the complexity of the *m* known LUT ($L(n)$) and proposed DLUT ($L_d(n)$), DCLUT ($L_{dc}(n)$), DNF-LUT ($L_{dnf}(n)$), $m=4$

DNF LUT is a competition about the number of variables is greater than 5, Fig. 22.

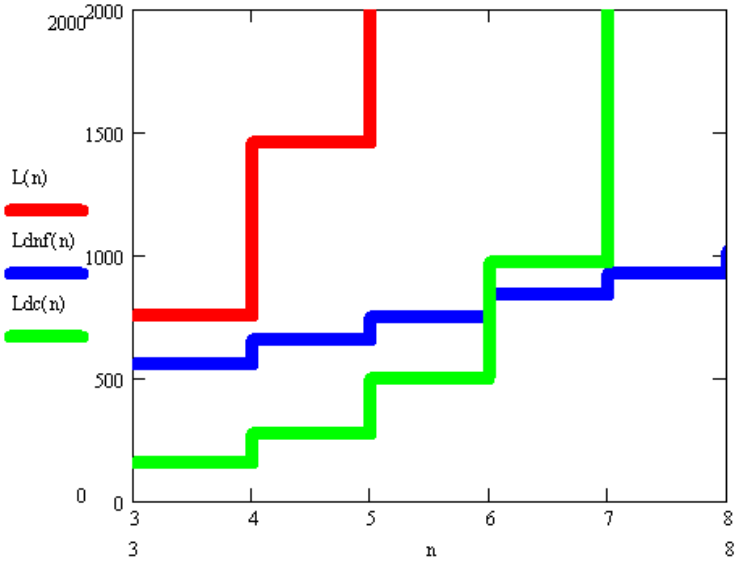

Figure 22. Comparison of the complexity of the *m* known LUT ($L(n)$) and proposed DLUT ($L_d(n)$), DCLUT ($L_{dc}(n)$), DNF-LUT ($L_{dnf}(n)$), $m=4; 3 \le n \le 8$

Comparison of the logical capacity (Number of transistors on a single function) of the *m* known LUT and proposed DCLUT, DNF-LUT presents Fig. 23, 24.

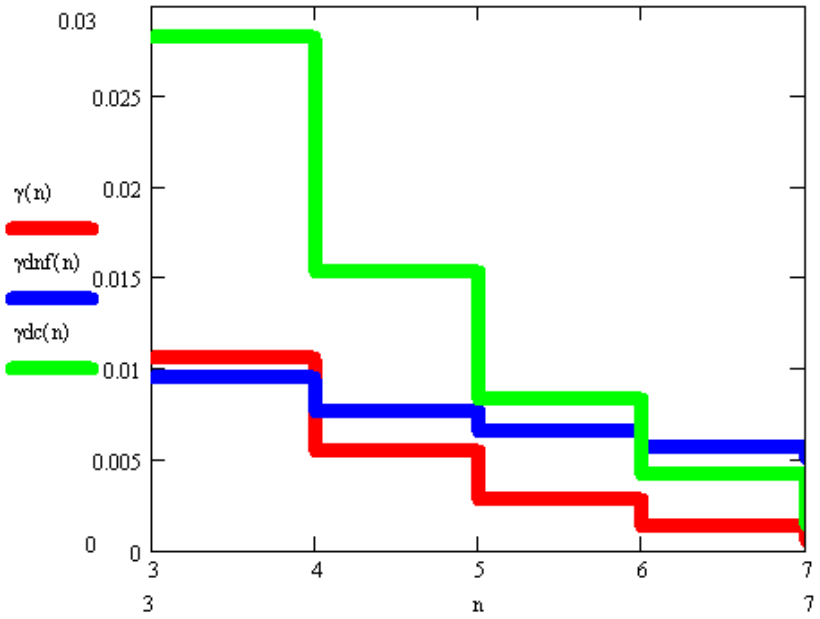

Figure 23. Comparison of the logical capacity of the *m* known LUT and proposed DCLUT, DNF-LUT, $m = 4$

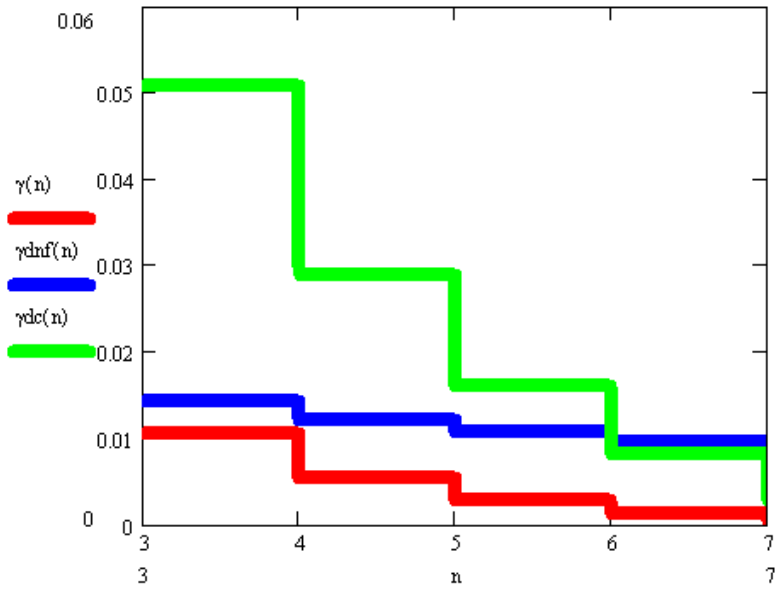

Figure 24. Comparison of the logical capacity of the *m* known LUT and proposed DCLUT, DNF-LUT, $m = 8$

IV. Conclusion

We analyze the complexity of the proposed technical solutions and the results of functional simulation. Proposed advanced LUTs for the logic systems significantly reduces hardware costs (from 10% to 60% and more - Fig. 20-23) without essential reducing performance. The simulation of the advanced LUTs executed in the NI Multisim 10 by National Instruments Electronics Workbench Group and confirmed the efficiency of patentable technical solutions. If necessary, a small number of calculating functions appropriate to use DLUT. With an average number of logic functions, it is advisable to use the DCLUT. However, a large

number of variables DNF-LUT has the best characteristics as the complexity of the other options is growing exponentially. In the future, it is advisable to consider the integrated use of a variety of solutions in a single FPGA and perform appropriate optimizations. One of the promising areas of research may be the creation of adaptive LUT-based proposed advanced LUT.

REFERENCES

[1] Tyurin S.F., Green Logic: Models, Methods, Algorithms. Green IT Engineering: Concepts, Models, Complex Systems Architectures,vol. 74, pp.69-86, 2017. DOI: 10.1007/978-3-319-44162-7

[2] Tyurin S.F., Green Logic: Green LUT FPGA Concepts, Models and Evaluations. Green IT Engineering: Components, Networks and Systems Implementation,vol. 105, pp.241-261, 2017. DOI: 10.1007/978-3-319-55595-9

[3] Drozd, A. et al.: The levels of target resources development in computer systems. In: Design & Test Symposium (EWDTS), 2014 East-West, pp. 1-5, IEEE, (2014).

[4] Mehta, Nikil. An ultra-low-energy, variation-tolerant FPGA architecture using component-specific mapping. Dissertation (Ph.D.), California Institute of Technology. (2013) Available at: http://thesis.library.caltech.edu/7226/1/Nikil-Mehta-2013.pdf.

[5] J. Cong, K. Minkovich. Optimality Study of Logic Synthesis for LUT-Based FPGAs. Available at: http://cadlab.cs.ucla.edu/~kirill/tcad06.pdf.

[6] Robert J. Francis A Tutorial on Logic Synthesis for Lookup-Table Based FPGAs. Available at: https://pdfs.semanticscholar.org/e879/aa350c38d381c41eeef0da95b739d2ede6c1.pdf.

[7] S. Brown. FPGA Architectura I Research: A Survey. Available at: http://arantxa.ii.uam.es/~die/%5BLectura%20FPGA%20Architecture%5D%20FPGA%20architectural%20research.%20A%20survey.pdf

[8] S. Brown, J. Rose. Architecture of FPGAs and CPLDs: A Tutorial. Available at: http://malun1.mala.bc.ca:8080/~pwalsh/teaching/355/Toronto_tutorial.pdf

[9] Tyurin S.F., Vikhorev R.V. Programmable logic device: patent RF №2573732; Published 27.01.2016, Bull. №3.

[10] Tyurin S.F. Programmable logic device: patent RF №2544750; Published 20.03.2015, Bull. №8.

[11] Conway, Lynn. Drafts of the Mead-Conway textbook, Introduction to VLSI Systems. University of Michigan. Retrieved 9 June 2015.