International Journal of Pure and Applied Mathematics
Volume 117 No. 22 2017, 143-147
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

Advanced FPGA Look Up Tables

Sergey Tyurin[1,2], Artem Grekov[3], Ruslan Vikhorev[1], Andrey Prokhorov[1]
[1]Perm National Research Polytechnic University, Perm, Russia
[2]Perm State National Research University, Perm, Russia
[3]Perm military Institute of National Guard Troops of the Russian Federation, Perm, Russia

Abstract - FPGA logic is 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 by means of the advanced LUTs in three main directions. The first direction is Double LUT (DLUT), which calculates two functions simultaneously with inactive transmission transistors subtree. The second direction is the decoder (DC LUT), which allows calculating the whole system of the logic functions by inversion of the tree of transistors. Such technique can significantly reduce hardware expenses for logic systems. The third direction is DNF-LUT, which allows the calculations of the system functions in disjunctive normal form (DNF) and more significantly reduces large number of variables LUTs hardware costs. Simulation of the proposed concepts was carried out 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: logic function; look up table; FPGA; perfect disjunctive canonical forms; comparison

1 Introduction

FPGA chips are widely used in computer technology [1]-[3]. There are quite many energy-saving methods of configuring FPGA [1],[2] for example, energy-efficient mapping and clocking, unused blocks power down and others. Ph.D. thesis [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 minimal energy/operation indicator. However, the expansion of the actual FPGA logic capacity for energy efficiency presented in the available sources is 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].

The objective goal of the paper is research and development of 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, which computes two and more functions in perfect disjunctive canonical forms (PDCF) simultaneously. Secondly, the LUT DNF technique is described by analogy with Prog the number of transistors to implement system of the logic functions in the known LUT and in all proposed LUTs is described.

2 Methodology

2.1 LUT FPGA Calculates Two Functions Simultaneously

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

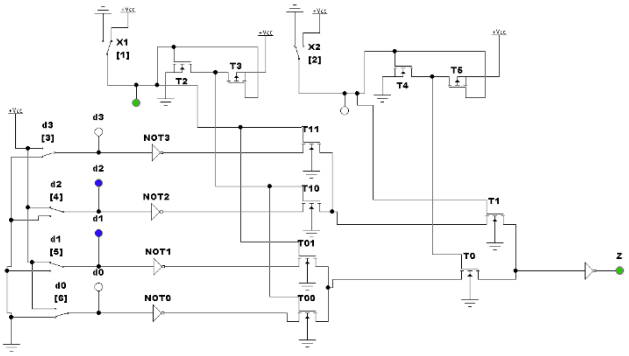


Figure 1. The 2-LUT model configured to calculate the exclusive OR function

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 only half the tree transistors is always activated (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 the 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. 3.

The simulation of the Double2-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 as presented in Fig. 2,4,5. To use the second half of the tree transistors additional transistors were introduced 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 simulates a constant connection – supply pins V_{cc} and Ground – "zero volts". Simulation confirms the

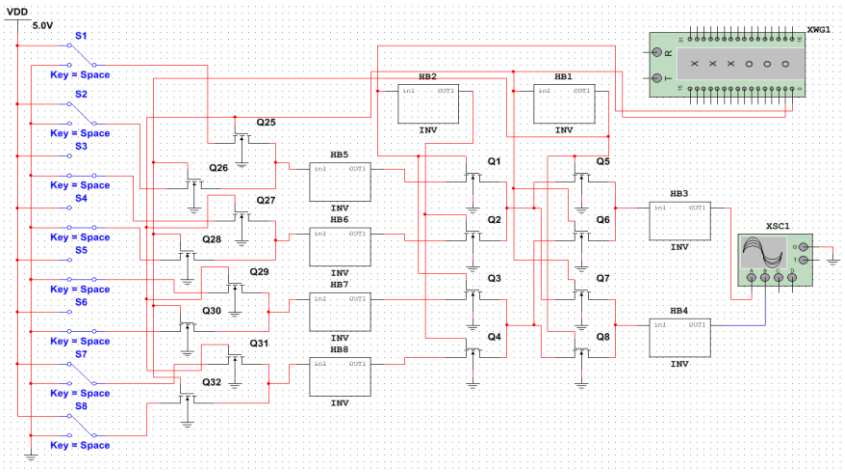


Figure 2. D2-LUT – dynamic model

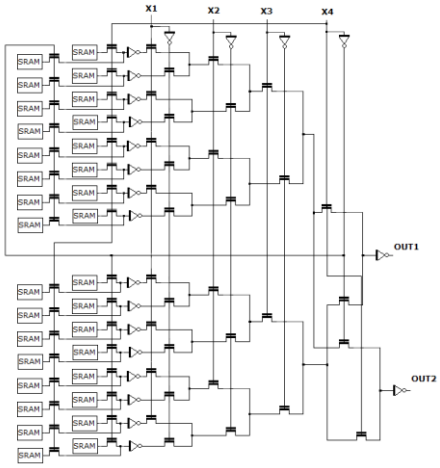


Figure 3. Double4-LUT (D4-LUT) calculates two functions simultaneously

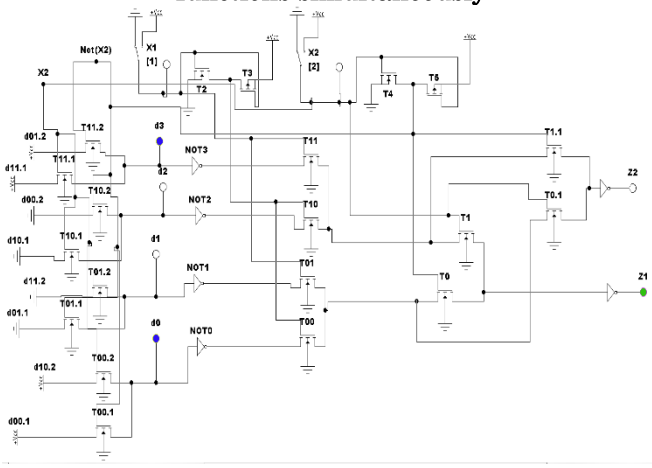
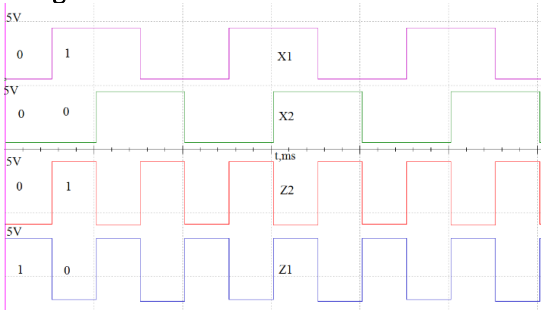


Figure 4. The simulation of the D2-LUT



2.2 DC LUT FPGA, Calculates System of the Logic Function

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

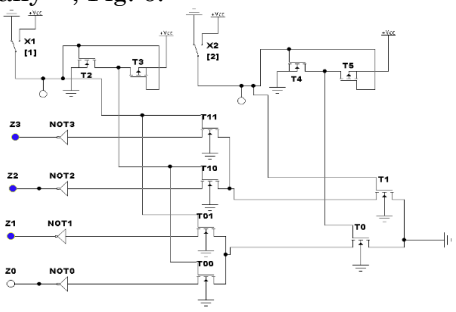


Figure 6. "Reverse" 2-LUT structure

In compliance with the design rules circuits of the transmission transistors required for a drain of each transistor T00, T01, T10, T11 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 presented in Fig. 7.

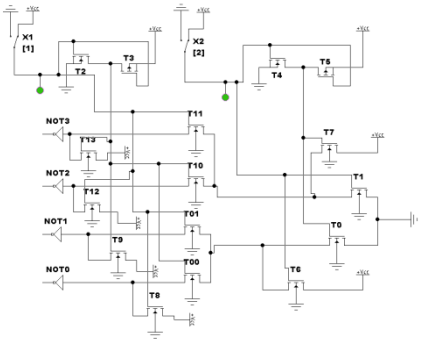


Figure 7. Reverse tree 2-LUT (DC 2-LUT) with alternate transistors T6, 7, 8, 9, 12, 13

Consider a 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 with one

programmable unit disjunctions configured to implement $x_1 \oplus x_2$ as shown in Fig. 8-10.

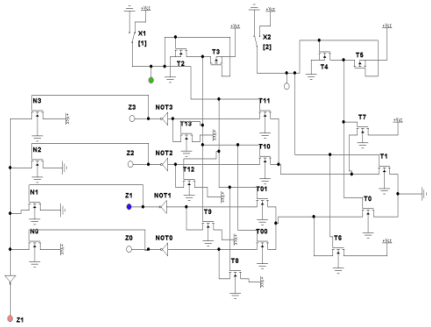


Figure 8. DC 2-LUT – static model

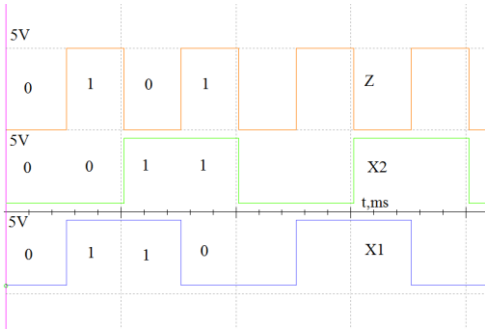


Figure 9. DC 2-LUT – waveform

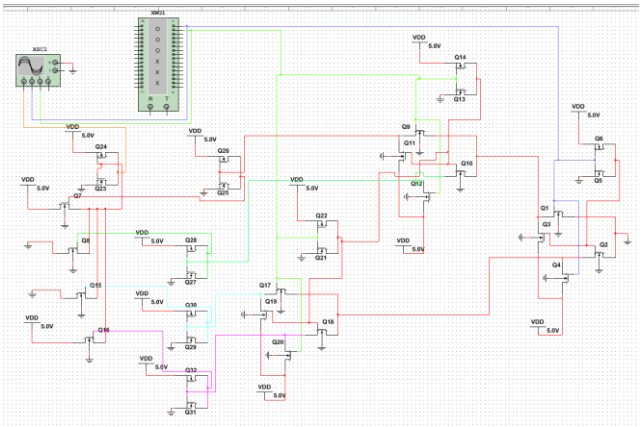


Figure 10. DC 2-LUT – dynamic model

2.3 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 in Fig. 11.

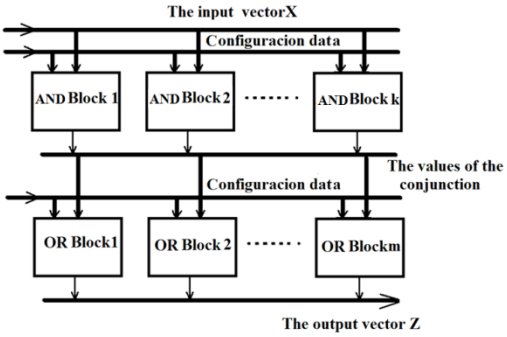


Figure 11. Architecture of DNF-LUT

Thus, instead of loading the truth table, it is loading only the programmed values of the length of the conjunctions n are loaded, where n – the number of variables of m logic functions. Occurrences of k conjunctions in m functions are also programmable tuning functions. For a given input set (vector n variables x) k AND blocks calculate the 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.

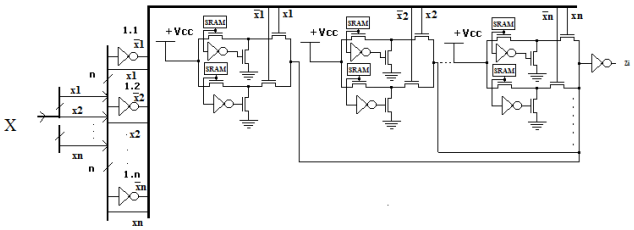


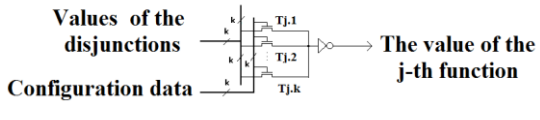
Figure 12. Architecture of the AND DNF-LUT block

One variable SRAM Setup is identified 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 "right" variable is active, 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 the "wrong" variable is active, inverters and additional transmission transistors are supply to the output logical zero. If all variables are "right" – the output z_i is a logical zero. The proposed structure of the block OR DNF-LUT is shown in Fig. 14.



block

A logical "1" at the output of the corresponding function will activate 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 is shown in Fig. 15.

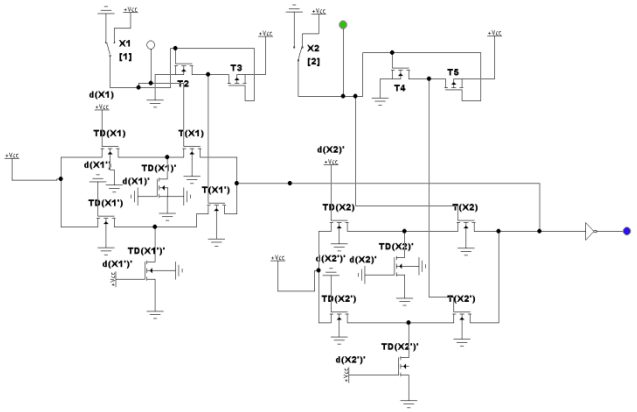


Figure 15. AND DNF-LUT Multisim model: $x_2x_1=0$, **since** $x_2=1, x_1=0$

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.

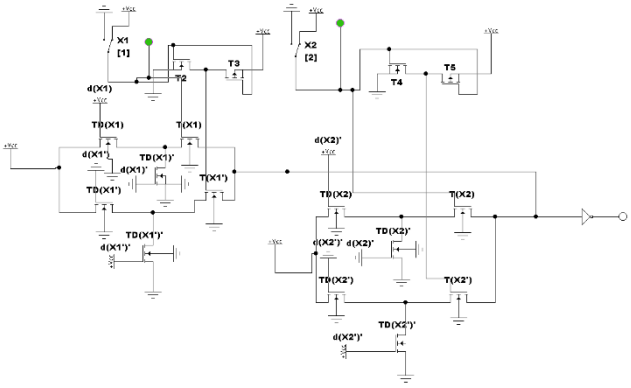


Figure 16. AND DNF-LUT Multisim model: $x_2x_1=1$, **since** $x_2=1, x_1=1$

3 Conclusion

An analysis of the complexity of the technical solutions and the results of functional modeling shows that the proposed advanced LUTs for the logic systems will significantly reduce hardware costs (from 10% to 60% and more – Fig. 20-23) without essential reducing of the performance. The simulation of the advanced LUTs executed in the NI Multisim 10 by National Instruments Electronics Workbench Group confirmed the efficiency of patentable technical solutions. If it is necessary to implement a small number of functions, it is preferable 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 have 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 optir may be the creation of an adaptive LUT based on the proposed advanced LUT.

References

Tyurin, S.F. "Green Logic: Green LUT FPGA Concepts, Models and Evaluations". Green IT Engineering: Components, Networks and Systems Implementation, 105, 241-261, 2017

Tyurin, S.F. "Green Logic: Models, Methods, Algorithms". Green IT Engineering: Concepts, Models, Complex Systems Architectures, 74, 69-86, 2017.

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.

Mehta, N. "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.

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

Robert, J. and Francis, A. "Tutorial on Logic Synthesis for Lookup-Table Based FPGAs". 1992. Available at: https://www.semanticscholar.org/paper/A-tutorial-on-logic-synthesis-for-lookup-table-bas-Francis/e879aa350c38d381c41eeef0da95b739d2ede6c1

Brown, S. "FPGA Architectura I Research: A Survey". 1996. Available at: http://arantxa.ii.uam.es/~die/%5BLectura%20FPGA%20Architecture%5D%20FPGA%20architectural%20research.%20A%20survey.pdf.

Brown, S. and Rose, J. "Architecture of FPGAs and CPLDs: A Tutorial". Available at: http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.pdf

Tyurin, S.F. and Vikhorev, R.V. "Programmable logic device: patent RF No. 2573732". Published 27.01.2016, Bull. No. 3, 2016.

Tyurin, S.F. "Programmable logic device: patent RF No. 2544750". Published 20.03.2015, Bull. No. 8, 2015.

Conway, L. "Drafts of the Mead-Conway textbook, Introduction to VLSI Systems". University of Michigan, 2015.

Grekov, A.V. and Tyurin, S.F. "FPGA logic element for implementation of disjunctive normal form". Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie, 60(6), 513-518, 2017.

Grekov, A.V. and Tyurin, S.F. "Fault tolerant logic cell FPGA". In Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017 IEEE Conference of Russian (pp. 287-290). IEEE.

Tyurin, S.F. and Grekov, A.V. "The decoding of LUT FPGA configuration of the finite state machine with Quartus II". International Journal of Applied 266, 2016.