Use of Natural LUT Redundancy to Improve Trustworthiness of FPGA Design

Alex Drozd, Miroslav Drozd, Mykola Kuznietsov

Institute of Computer Systems, Odessa National Polytechnic University,
1 Shevchenko St., 65044 Odessa, Ukraine

Drozd@ukr.net, miroslav_dr@mail.ru,
koliaodessa@mail.ru

Abstract. Paper is devoted to a design FPGA problem regarding increase in trustworthiness of the results calculated in digital components of computer systems. The modern CAD solves this problem by implementation of significant hardware redundancy in fault tolerant decisions. The method for increase in trustworthiness of the results calculated on FPGA with the LUT-oriented architecture is offered. This method is directed to improving of the ready project without change of its hardware decision. The method generates 16 versions in programming of LUT memory and suggests selecting the best version by the given criterion. The method is shown on the example of masking of short circuit between LUT inputs. The method allows selecting the decision taking into account risk-value of bits in LUT memory.

Keywords. FPGA design, resource-based approach, LUT-oriented architecture, natural redundancy, trustworthiness of results, masking of short circuit

Key Terms. High Performance Computing, Concurrent Computation, Model, Method, Simulation

1 Introduction

We offer a method for improving of FPGA projects in view of ample opportunities and the perspective directions in design of computer systems and their components on FPGA. External manifestations of these two aspects are watched in a gain of the market in the field of the design of systems for digital processing of signal and images, accelerators, reconfigurable systems and safety-related instrumentation and control systems [1 – 4].

The internal reasons of successful development of FPGA design can be considered by means of resource-based approach which analysis process of integration of the artificial world created by the human into the natural world (NW). It allows not only to explain history of the computer world which repeats in development the NW during a short time but also to offer new methods based on systematization in features and perspectives of further development [5].

Integration process consists in the solution of challenges of the NW, including problems of the computer world. The solution is based on achievement of throughput (for execution of all amount of works for limited time), of result trustworthiness and on attachment of resources: models, methods and means.

The models are human understandings of the NW and its components. Methods serve for conversion and an assessment of resources. Models and methods form an informational part of the resources but means (materials and tools) belong to the technological one. Resources develop from simple to real by structuring under features of the NW. Simple forms are exact and sequential according to initial opportunities and understandings of the human. Real resources reflect features of the NW [6].

Development of computer resources most brightly showed two such features: parallelism and fuzziness [7, 8]. In the resource development occurring under these features, the resource-based approach identifies a number of levels: replication, diversification and autonomy. In the hierarchy they occupy the low, L, the middle, M and the high, H levels, respectively. Each preceding level serves for the next one. In all the levels, from L to H, the goal is surviving, i.e. integration into the NW. It is provided by different methods: increasing throughput, trustworthiness of results and access to resources (self-sufficiency), respectively [9].

The LUT-oriented architecture of FPGA projects has the high level of fuzziness in the form of natural redundancy in the organization and functionality. We offer a method which shows opportunities to use this redundancy for development of a CAD in the modes of FPGA-project optimization for improving of result trustworthiness.

The method allows replicate different versions of the LUT program code for the ready FPGA project with the subsequent choice of the best by the selected criterion. The method is shown on the example of fault masking for the purpose of increase in trustworthiness of the results calculated on FPGA. The problem of FPGA project improvement by criterion of result trustworthiness is defined in section 2. Section 3 describes a model of data transfer between LUT. Versions of LUT programming are generated in section 3. In section 4 the received versions are evaluated from a line item of masking of faults like shorts circuit of adjacent inputs of the LUT. Increase in trustworthiness of the results calculated on FPGA in case of faults like shorts is shown in section 5. Experimental results are considered in section 6. The received results are analyzed in the conclusions.

2 Problem Definition

All levels of resource development are represented in FPGA design. The matrix and pipeline types of parallelism corresponding to levels of replication and diversification are supported by arrays of the configurable blocks and the structure of a logical element as section of the pipeline containing the LUT and the register [10].

Matrix parallelism restricts possibilities in paralleling of computations by data dependency when high orders of numbers will be processed after low orders, and by control dependency with waiting for computation of conditions in branching of an algorithm. Pipeline parallelism removes data dependency processing at the same time both high and low orders in different sections of the pipeline [11].

A method of results preparation belongs to the high level of autonomy. This method is free from both restrictions, calculating a set of possible results, not waiting for receiving all operands [12].

All the libraries, including software and library of IP cores used in CAD are built by the method of result preparing. All of the modern design of digital systems and their components are also based on the method of result preparing. For example, the FPGA chips are preparations for a lot of projects, and the chip programmed for a specific project contains the prepared results in the tables stored in the LUT (Look-Up Table) memory [13].

Mainly due to the preparation of the results, FPGA projects receive features, allowing provide high level a number of characteristics: capacity of calculations and the trustworthiness of their results, versatility in functionality, efficiency of design, manufacturability and flexibility [14].

However, the modern CAD is oriented first of all on matrix parallelism. The FPGA chips contain sets of the prepared iterative array multipliers and the prepared circuits of look-ahead carry for design of digital units with matrix parallelism [15].

As a rule, the CAD supports the optimization modes in energy consumption and throughput, ignoring requirements for the result trustworthiness important for the safe computer systems including autonomous systems relating to levels of diversification and autonomy [16, 17]. Opportunities of such systems completely are defined at design stage, for example, onboard equipment of space crafts for distant flights, the pacemakers implanted in a body of the person.

Possibilities of FPGA design associated with natural redundancy of the LUT-oriented architecture allow improve significantly projects by a method of versions which repeats development of the NW by means of mutations. This method provides replication of the ready decision with deviations, admissible from it, in any parameters with the subsequent choice of decision which is the best by the main criterion. The choice of the decision can be executed with use of analytical estimates or simulation [12].

Use of the method of versions completely comes true within the principle of ALARA (As Low As Reasonability Applicable/Practicable), when there is no alternative of improving of the main characteristics of the decision directed to the greatest possible lowering of risk at the expense of really available limited resources [18].

Thus, the problem, motivation of its decision and resources necessary for this purpose are as follows:

1. As a rule, the modern CAD doesn't support the modes of project optimization in trustworthiness of results.

2. Increase of result trustworthiness is an important condition of maintenance in the functional safety of computer systems in areas of critical application and is traditionally provided with use of fault-tolerant decisions with significant hardware redundancy.

3. FPGA projects have the high level of fuzziness which allows to use effectively the method of versions for increase of result trustworthiness on the basis of natural redundancy of the LUT-oriented architecture without extra hardware.

We offer the method using natural redundancy to replicate different versions of the program LUT code for the given hardware implementation of the FPGA project. Replication is executed with changes, unessential for project parameters. The choice of the version is carried out in the most important parameter.

The method is shown on the example of the best masking of typical faults of the digital circuits for the purpose of the greatest increase of trustworthiness in results of the computation executed on FPGA.

3 The Model of Data Transfer between LUT

We can consider the LUT-oriented architecture on the example of Cyclone FPGA Family Data Sheet of Altera Corporation. Cyclone structure contains arrays of logic elements. Each logic element consists of a four-input LUT and programmable register. The LUT is a function generator that can implement any function of four variables *A*, *B*, *C* and *D*. Inputs of the LUT are called as well as variables [19].

We offer model of data transfer between LUT which allows change a program code for LUT without changing the hardware structure of FPGA-project.

The program code can be diversely replicated with use of two versions in programming of ordered pair of the LUT units connected among themselves. The output of the first of them is connected directly or via the programmable register (or other elements with implementation of a self-dual function) to an input of the second LUT. Two versions of programming which differ from each other in direct or inverse value of the bit transferred from the first LUT to the second one are shown in Fig. 1.

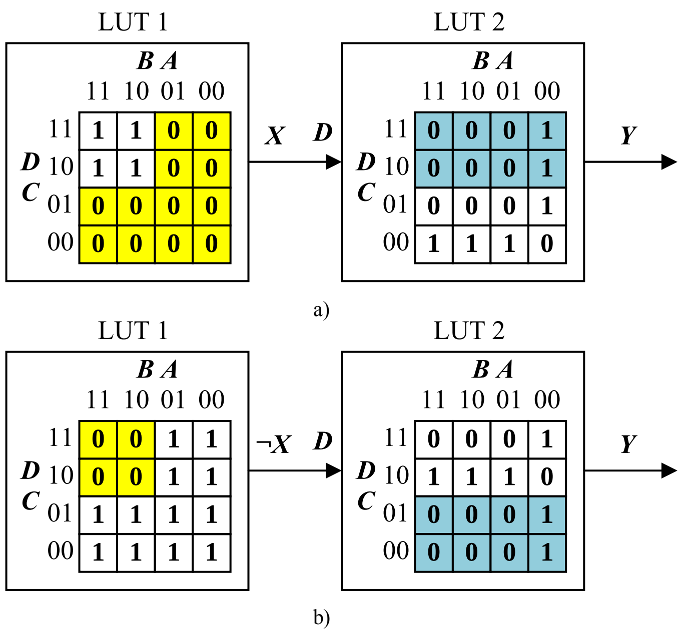


Fig. 1. Two versions of LUT units programming: a – transfer of a direct value of *X*; b – transfer of an inverse value of *X*

The first LUT 1 and second LUT 2 generate functions $X = D \wedge B$ and $Y = (A \wedge B) \oplus (C \wedge D)$, respectively. The functions are described by tables which rows and columns are determined by values of 2-bit codes *DC* and *BA*, respectively. These codes are made of the *A*, *B*, *C*, *D* variables. Connection between units transfers value *X* from an output of the LUT 1 to the *D* input of the LUT 2 (Fig. 1, a).

The second version in programming of units can be executed by transmission of inverse value $\neg X$ between them (Fig. 1, b). In this case, all bits of the LUT 1 will be inverted. For saving value *Y*, i.e. for compensating of inverse value *X*, the table of the LUT 2 changes mutual position of values "0" and "1". In case of inverse value of the variable *D*, positions of two top rows and two bottom ones are interchanged.

4 Versions in Programming of the LUT Units

Direct and inverse values of the variables *A*, *B*, *C*, *D* generate 16 different versions in programming of the LUT 2. All of them are shown in Fig. 2.

	$B\ A$				$B\ \neg A$				$\neg B\ A$				$\neg B\ \neg A$			
	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
D	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
C	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1
	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
D	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
$\neg C$	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1
	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
$\neg D$	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1
C	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1
$\neg D$	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
$\neg C$	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0

Fig. 2. Versions in programming of the LUT 2

All versions can be received on condition of connection of all four inputs of the LUT 2 to outputs of four units as LUT 1.

The FPGA-project containing *n* pair of LUT can be replicated into a set of 2^n decisions which save uniform hardware structure and differ in a program code. For example, the 10 pair of LUT provides 1024 versions of the project. These versions can significantly differ in a number of parameters, including trustworthiness of the calculated results.

5 Increase in Trustworthiness of the Results Calculated on FPGA

The offered method can be shown on the example of increase in trustworthiness of the results calculated on FPGA in case of faults like shorts between adjacent inputs of the LUT [20]. Tables for LUT 2 in cases of the correct functioning and short circuit *A*↔*B*, *B*↔*C*, *C*↔*D* between inputs *A* and *B*, *B* and *C*, *C* and *D* are shown in Fig. 3.

	B A				***A↔B***				***B↔C***				***C↔D***			
A	0	0	0	1	0	1	1	1	0	0	0	1	0	0	0	1
B	0	0	0	1	0	1	1	1	0	1	0	1	1	1	1	0
C	0	0	0	1	0	1	1	1	0	0	1	0	1	1	1	0
D	1	1	1	0	1	0	0	0	1	0	1	0	1	1	1	0

a)

	B A				***A↔B***				***B↔C***				***C↔D***			
¬A	0	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1
¬B	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
¬C	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
¬D	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

b)

Fig. 3. Tables for cases of the correct and faulty functioning of the LUT 2: a – the initial programming of the LUT; b – programming of the LUT with inverse values of inputs

Short circuit of two inputs distorts their values "01" and "10", substituting with them by "00" value. It leads to substitution of eight bits determined in the table by "01" and "10" values with four bits addressed on "00" value. Comparing of tables for cases of the correct and faulty functioning of the LUT 2 allows detect the erroneous bits highlighted in the color.

The first row of tables is shown for the initial programming of the LUT 2 (Fig. 3, a). Short circuit *A*↔*B* determines 8 erroneous bits in two internal columns of the table. Short circuit *B*↔*C* is shown in four erroneous bits. Short circuit *C*↔*D* determines 8 erroneous bits in two internal rows of the table. The total quantity of erroneous bits is equal 20. It makes 42% of total quantity of bits in memory of LUT.

The second row of tables is shown for the programming of the LUT 2 with inverse values of inputs *A*, *B*, *C* and *D* (Fig. 3, b). Short circuits A↔B and C↔D don't lead to distortion of bits in the table. Short circuit *B*↔*C* is shown in four erroneous bits. The total quantity of erroneous bits is equal 4. It makes 8% of total quantity of bits in memory of LUT. Thus, conversion of the version with the initial programming of the LUT units reduces the number of erroneous bits by 5 times, i.e. repeatedly improves masking of short circuits and increases trustworthiness of the results calculated on FPGA. This effect is reached without any changes in the hardware implementation of the circuit decision, i.e. after all types of its optimization. The advantage of the offered method is the independent analysis of the LUT pairs that significantly reduces time necessary for a choice of the best version of the FPGA project.

The method is shown for a case of identical risk-values of erroneous bits for obtaining authentic results. But these values can change depending on the frequency of use of the bit and consequences of its distortion. The accounting of these two factors makes a method as risk-oriented. Such development of the method includes two steps. On the first step, the frequency of the appeal to each bit of each LUT should be evaluated. This step can be executed by simulation of operation of the circuit on characteristic input data. The program model of the circuit decision allows to count the number of appeals to each bit of memory and to calculate the frequency of its use. The second step requires expert estimates of consequences for every error.

6 Experimental Results

The experimental verification of the offered method was executed in process of preparation of a lab classes within the project TEMPUS GREENCO "Green Computing & Communications" (530270-TEMPUS-1-2012-1-UK-TEMPUS-JPCR) [21].

The program model is developed for the analysis of LUT pairs using Delphi 10 Seattle demo-version [22]. The analysis is made in two modes:

- for a specific case of programming of the LUT 2 in view of the risk-value of each bit for trustworthiness of result;
- for all 216 functions generated in the LUT 2 on condition of identical value of bits in relation to trustworthiness of result.

The main panel of the program with setting of the risk-values of bits in LUT 2 memory for function *Y* is shown in Fig. 4.

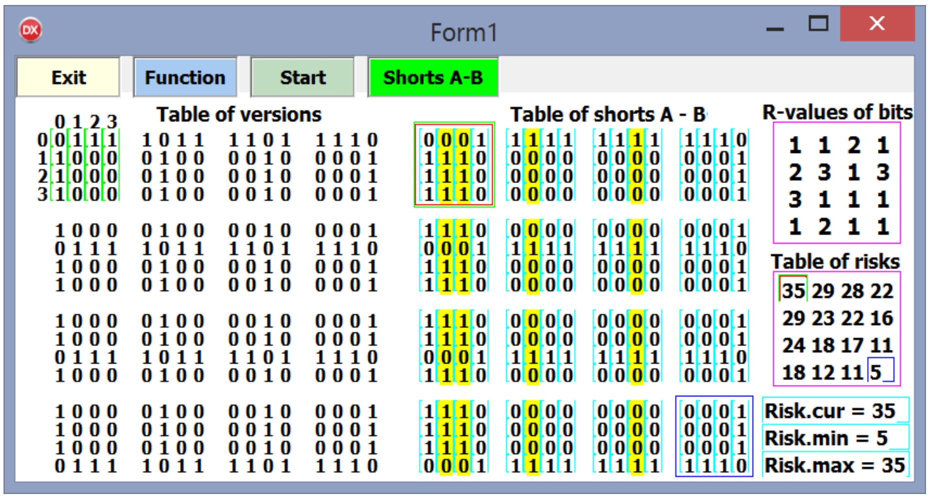
The program calculates risk for each of 16 versions in programming of the LUT unit according to the given table “R-values of bits” of bit risk-values and sets of the erroneous bits in versions. For example, in case of table “R-values of bits”, the table of risks for 16 versions shows that the risk Risk.cur = 35 of the initial version matches maximum Risk.max = 35 and exceeds the minimum risk Risk.min = 5 by 7 times.

Thus, reprogramming of LUT unit can repeatedly reduce the risks associated with faults in implementation of FPGA projects. Viewing of all functions generated in the LUT 2 unit showed that the average scoring of the best version in comparison with current one and the worst is equal 21% and 53% of the masked errors, respectively.

7 Conclusions

LUT-oriented architecture of FPGA-projects was widely used in various areas of information technologies owing to the high levels of development of its resources:

models, methods and means. At the same time, its opportunities are used not fully, saving reserves for the best application.



```
         Table of versions                   Table of shorts A - B        R-values of bits
  0 1 2 3
0 0 1 1 1   1 0 1 1   1 1 0 1   1 1 1 0    0 0 0 1   1 1 1 1   1 1 1 1   1 1 1 0    1 1 2 1
1 1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1    2 3 1 3
2 1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1    3 1 1 1
3 1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1    1 2 1 1

  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1    Table of risks
  0 1 1 1   1 0 1 1   1 1 0 1   1 1 1 0    0 0 0 1   1 1 1 1   1 1 1 1   1 1 1 0    35 29 28 22
  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1
  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1    29 23 22 16

  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1    24 18 17 11
  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1
  0 1 1 1   1 0 1 1   1 1 0 1   1 1 1 0    0 0 0 1   1 1 1 1   1 1 1 1   1 1 1 0    18 12 11 5
  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1

  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1    Risk.cur = 35
  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1    Risk.min = 5
  1 0 0 0   0 1 0 0   0 0 1 0   0 0 0 1    1 1 1 0   0 0 0 0   0 0 0 0   0 0 0 1
  0 1 1 1   1 0 1 1   1 1 0 1   1 1 1 0    0 0 0 1   1 1 1 1   1 1 1 1   1 1 1 0    Risk.max = 35
```

Fig. 4. Panel of the program with setting of the risk-values of LUT memory bits

All levels of resource development are provided to FPGA chips:

- replication in matrix parallelism which is offered for design of digital units with data processing in parallel codes;
- diversification in the pipeline parallelism supported in structure of logical elements or reconfigurable logic blocks;
- autonomy in preparation of results and circuits of their distribution in adding units, in preparation of iterative array multipliers.

However, the matrix parallelism relating to the low level of resource development dominates. For example, preparation of results is used for improving of circuit decisions with matrix parallelism. The modern CAD offers the optimization modes aimed at enhancement of FPGA-projects in parameters of throughput and energy consumption which are important at the level of replication and autonomy, respectively. At the same time, improving of projects in trustworthiness of the results calculated on FPGA isn't executed. However increase of trustworthiness is important for safety-related computer systems including such autonomous systems.

The offered method allows improve ready FPGA-project without change of the reached throughput and energy consumption. The method saves the hardware decision and executes reprogramming of LUT, each of which can have 16 versions of programming. The method is shown on the example of the best masking of short circuit between LUT inputs. Reprogramming of LUT allows reducing quantity of possible errors by 5 times. Development of the method taking into account risk-value of bits in LUT memory can raise this index considerably.

References

1. Bakhmach, E., Kharchenko, V., Siora, A., Sklyar, V., Tokarev, V.: Design and Qualification of I&C Systems on the Basis of FPGA Technologies. In: 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT 2010), pp. 916--924. Las Vegas, Nevada (2010)
2. Dick, C. H., Harris, F. J.: Direct Digital Synthesis – Some Options for FPGA Implementation. In: SPIE International Symposium on Voice Video and Data Communication: Reconfigurable Technology: FPGAs for Computing and Applications, pp. 2-10. Stream, Boston, MA, USA (1999)
3. Palagin, A.V., Opanasenko, V.N.: Design and application of the PLD-based reconfigurable devices. Design of Digital Systems and Devices, Springer Verlag, Berlin, Heidelberg, vol. 79, pp. 59--91 (2011)
4. Melnyk, A. O., Melnyk, V. A.: Personal Supercomputers: Architecture, Design, Application: Lvivska Polytechnika, Lviv, Ukraine (2013)
5. Drozd, J., Drozd, A., Antoshchuk, S., Kharchenko, V.: Natural Development of the Resources in Design and Testing of the Computer Systems and their Components. In: Proc. 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 233--237. Berlin, Germany (2013)
6. Tanenbaum, A.: Structured Computer Organization, 4th ed. Prentice Hall, Upper Saddle River, NJ (1999)
7. Kondratenko, G., Kondratenko, Y., Romanov, D.: Fuzzy Models for Capacitative Vehicle Routing Problems in Uncertainty. In: Proc. International DAAAM Symposium, pp. 205--206 (2006)
8. Drozd J., Drozd A.: Models, Methods and Means as Resources for Solving Challenges in Co-Design and Testing of Computer Systems and their Components. In: Proc. 9th International Conference on Digital Technologies (DT'2013), pp. 176--180. Zhilina, Slovakia (2013)
9. Drozd, J., Drozd, A., Maevsky, D., Shapa, L.: The Levels of Target Resources Development in Computer Systems. In: Proc. IEEE East-West Design & Test Symposium, pp. 185—189. Kiev, Ukraine (2014)
10. Drozd, J., Drozd, A., Antoshchuk, S., Kushnerov, A., Nikul, V.: Effectiveness of Matrix and Pipeline FPGA-Based Arithmetic Components of Safety-Related Systems. In: Proc. 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. pp. 785--789. Warsaw, Poland (2015)
11. Kharchenko V.S. (ed): Green IT-Engineering. Volume 1. Principles, Models, Components. National Aerospace University "KhAI", Kharkiv, Ukraine (2014)
12. Drozd A., Kharchenko, V. (eds): Online Testing of the Safe Instrumentation and Control Systems. National Aerospace University named after N.E. Zhukovsky "KhAI", Kharkiv, Ukraine (2012)
13. Netlist Optimizations and Physical Synthesis. Qii520072.0. Quartus II Handbook. Vol. 2: Altera Corporation (2004)
14. Kharchenko, V.S., Sklyar, V.V. (eds): FPGA-based NPP I&C Systems: Development and Safety Assessment. RPC Radiy, National Aerospace University "KhAI", SSTC on Nuclear and Radiation Safety, Kharkiv, Ukraine (2008)
15. Xilinx Staff. Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet (2005)
16. PowerPlay Power Analysis. Qii53013. Quartus II Handbook: Altera Corporation (2013)
17. Design Optimization for Altera Devices. Qii52005-2.0. Quartus II Handbook. Vol. 2: Altera Corporation (2004)

18. Kharchenko, V.S. (ed): Safety of Critical Infrastructures: Mathematical and Engineering Methods of Analysis and Ensuring: Ministry of Education and Science of Ukraine, National Aerospace University "KhAI", Kharkiv, Ukraine (2013)
19. Cyclone FPGA Family Data Sheet. Altera Corporation. Online: http://www.altera.com (2003)
20. Blyzniuk, M., Kazymyra, I., Kuzmicz, W.: Probabilistic Analysis of CMOS Physical Defects in VLSI Circuits for Test Coverage Improvement. Microelectronics Reliability, 41(12), 2023--2040 (2001)
21. Drozd, A., Kharchenko, V. (eds): Green Harware and Programmable Systems: Ministry of Education and Science of Ukraine, National Aerospace University "KhAI", Kharkiv, Ukraine (2015)
22. Delphi 10 Seattle: Embarcadero. Online: https://www.embarcadero.com/ru/products/delphi (2015)