Improving of a Circuit Checkability and Trustworthiness of Data Processing Results in LUT-based FPGA Components of Safety-Related Systems

Oleksandr Drozd, Miroslav Drozd, Oleksandr Martynyuk, Mykola Kuznietsov
Institute of Computer Systems, Odessa National Polytechnic University,
ave Shevchenko 1, 65044 Odessa, Ukraine
drozd@ukr.net, miroslav_dr@mail.ru, anmartynyuk@ukr.net,
koliaodessa@mail.ru

Abstract. The possibility of improving on attributes of a solution which are traditionally opposed each other is proved. For digital components of safety-related systems, the method of improving on attributes of a checkability of the circuit and trustworthiness of the results calculated on FPGA with the LUT-oriented architecture is offered. The method is directed to improving of the ready project by a choice of the version of a program code without change of the hardware decision. Versions of LUT memory programming and a set of faults on which these versions exert impact are generated. Faults of shorts between adjacent address inputs of LUT are considered. Operation of the circuit is simulated on all versions and on all set of faults. The method selects the versions providing increase in a checkability of the circuit in a normal mode and trustworthiness of results in emergency mode of the safety-related systems.

Keywords: safety-related system, digital component, FPGA, LUT-oriented architecture, checkability of the circuit, trustworthiness of the results

Key terms: HighPerformanceComputing, ConcurrentComputation, Model, Method, Simulation

1 Introduction

Objects of the increased risk in power engineering, on transport and other areas became natural surroundings of the human. Power stations and the high-speed railroad, aircraft permanently increase complexity and capacities. It requires the appropriate enhancement of the safety-related systems ensuring the functional safety of these objects [1, 2].

We offer a method for improving the digital components of safety-related systems in two attributes important for the functional safety: checkability of the circuit and trustworthiness of results. The insufficient checkability of the circuit restricts possibilities of on-line testing to detect faults in the course of computation. This creates a problem of the hidden faults in safety-related systems [3, 4]. Trustworthiness of results plays a crucial role for accident prevention and lowering of their consequences. Both attributes are traditionally opposed each other, i.e. improving of one attribute will be reached due to deterioration in another. The high ckeckability of the digital circuit reduces trustworthiness of result as promotes frequent manifestation of faults in the form of errors. The low checkability of the circuit increases trustworthiness of results as it masks faults better. High trustworthiness of results is provided by development of fault-tolerant circuits [5], structural redundancy of which reduces a checkability.

Such opposition of these and other attributes is perceived as a norm, however it has no objective character, and it is only result of limitation of our models, i.e. our ideas concerning the nature of objects of a research. A certain step in development of models can be taken, using resource-based approach [6, 7]. According to this approach, the human develops models, methods and means, component resources, from simple to real, by passing in the consciousness the levels of resources development: replication, diversification and autonomy. Simple forms of resources are exact and sequential according to initial representations and opportunities of the human. Real resources reflect the natural world in which information technologies most show two features: parallelism and fuzziness. All history of development of the computers is the evidence of increase in level of parallelism and fuzziness of decisions, i.e. it is an example of structuring under features of the natural world [8, 9]. Use of exact models and methods is restricted by need of solving problems in the conditions of uncertainty [10, 11].

One of fuzziness forms is redundancy shown in a set of versions of a solution. Such diversity is supported by design of digital circuits on a programmable logic which removes a contradiction between universality and specialization of an circuit element basis: the universalization is provided with stamping of chips, and specialization – programming of their functions [12, 13]. These and other advantages of programmable chips became a basis for use of FPGA in case of digital component design for safety-related systems [14, 15]. Creation of versions is provided in FPGA projects with use of hardware reconfiguration technologies [16, 17]. However versions of FPGA projects can also be created only by change of a program code in case of full save of the project hardware [18]. We offer a method which uses version redundancy of FPGA projects with the LUT-oriented architecture [18] for a choice of the most effective versions of a code of programming in digital components of safety-related systems. The choice is executed by criterion of the best attributes of a checkability of the circuit and trustworthiness of the results calculated on this circuit. The method analyzes faults of shorts between adjacent address inputs of LUT. Faults of shorts concerns to the most typical for CMOS of integrated circuits [19].

Basic ideas of a method are described in section 2. The possibility of elimination of a traditional contradiction between a checkability of the circuit and trustworthiness of the results calculated by this circuit is considered. Influence of versions on faults of shorts between adjacent address inputs of LUT is analyzed. An opportunity and feasibility of the maximum distinguishing of attributes of a circuit checkability and trustworthiness of the results is shown. Program implementation of a method is provided in section 3. Basic data and steps of the program execution including simulation of operation of the circuit are defined. The results of simulation allowing to select the version of the circuit programming are shown in the form of the table.

2 Basic Ideas of a Method

The main position of the offered method is the possibility of elimination of a traditional contradiction between a checkability of the circuit and trustworthiness of the results calculated by this circuit. In other words, the method is based on a possibility of development of the methods with multiple effect [20]. These methods are aimed at simultaneous improving of attributes which are traditionally opposed each other. What such tradition is based on? Why a norm is the method of the "Robbing Peter to pay Paul" where one attributes improve due to deterioration in others? Whether the multiple effect, i.e. improving of several attributes without loss in others is possible?

Responses to these questions should be looked for in consciousness of the human. His limited models restrict our opportunities. The human thinks by exact data, i.e. integers by the nature. These are numbered data. In the consciousness the human creates a set of any objects and numbers its elements with abstracting from the features distinguishing real objects. It promotes perception of the natural world at the level of replication where the method of the "Robbing Peter to pay Paul" dominates. The modern computer world, generally is at this bottom level of development – replication. Array structures in computer systems and their hardware components are stamped using identical operational elements. Software modules are also stamped for the purpose of their connection to different applications [21, 22]. Additionally stamped elements allow to increase throughput instead of deterioration in an index of complexity.

At the same time, the natural world doesn't create anything identical, and even stamping generates versions, i.e. higher level of diversification in resources development is implemented. In particular, the increase in trustworthiness based on the double account or use of majority systems is reached not due to stamping of the duplicating elements, and thanks to their not identity, i.e. version redundancy. Systems of critical application are development of computer systems with diversification of an operating mode by its division on normal and emergency.

Substitution in our consciousness of actual level of diversification on the level of replication distorts perception of methods of resources development: methods with the multiple effect are perceived one-sidedly as methods of the "Robbing Peter to pay Paul", tabooing overcoming traditional contradictions between attributes of the problem solution. Being in the real natural world, the human can't but develop methods of the multiple effect, but does it unconsciously and uses their results partially or blindly, without guessing this use. The way to use of methods with the multiple effect lies through diversification, i.e. distinguishing of versions.

Diversification of an operating mode of safety-related systems by its division into normal and emergency allows to diversify requirements to a checkability of the circuit and trustworthiness of results in relation to the normal and emergency mode. The checkability of the circuit shall be maximum and minimum respectively in normal and emergency modes. In a normal mode it provides the best conditions for fault detection with methods of on-line testing, and in emergency mode it restricts action of faults. Trustworthiness of results shall be minimum and maximum respectively in normal and emergency modes. Low trustworthiness of results in a normal mode accompanies on-line testing in detection of faults which are found by identification of non-authentic results. High trustworthiness of results is an important condition of successful actions in an emergency mode of safety-related system. Thus, opposition of a checkability of the circuit and trustworthiness of results changes at the level of diversification on their coherence in the direction of support in the functional safety of safety-related systems.

The following position of a method is aimed at the maximum distinction of attributes of a checkability of the circuit and trustworthiness of results in order to direct of them to opposite extremal values. This position is based on use of versions in a program code for the LUT-oriented architecture of FPGA projects and versions in manifestation of faults like shorts. Creation of versions in a program code for the LUT-oriented architecture of FPGA projects was considered in [19] for each pair of LUT 1 and LUT 2 if the output of LUT 1 is connected to an address input of LUT 2. The signal from an output of LUT 1 can be transmitted to an address input of LUT 2 by direct or inverse value based on two versions in programming of pair of LUT 1 and LUT 2. The second version differs from the first one in inverting of all bits of memory of LUT 1 for inverting of its output and in replacement of bits in memory of LUT 2 for compensating of inverse on its address input. Each pair of LUT 1 and LUT 2 generates 2 versions irrespective of other such pairs. Versions of manifestation of faults like shorts of two adjacent address inputs of LUT are shown in Fig. 1.

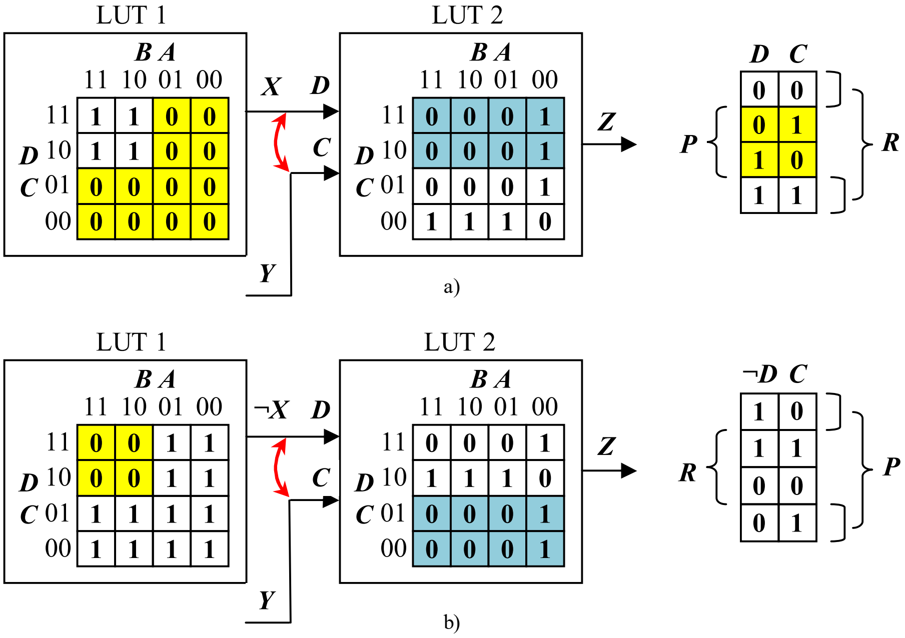


Fig. 1. Areas E of fault manifestation in two versions of LUT units programming: a – area P for a direct value of X; b – area P for an inverse value of X

In the left of Fig. 1, two versions in programming of pair of LUT 1 and LUT 2 in the conditions of shorts of address inputs D and C of LUT 2 are shown.

In the first version, the LUT 1 and LUT 2 generate functions $X = D \wedge B$ and $Z = (A \wedge B) \oplus (C \wedge D)$, respectively. These functions are represented in tables which rows and columns are determined by values of 2-b codes DC and BA, respectively. The direct value of X is calculated at an output of LUT 1 and arrives at a direct address input D of the LUT 2. The signal Y arrives at an input C of the LUT 2 which forms signal Z at an output. The second version calculates the same result Z at the output of LUT 2 by using of functions $\neg X$ and $Z^* = (A \wedge B) \oplus (C \wedge \neg D)$, where $\neg D = \neg X$. Inverse value of X

is calculated at an output of LUT 1 and arrives at an inverse address input D of the LUT 2. Inverse of an address input of D is provided with replacement of bits in memory of LUT 2: two top rows of memory bits are interchanged with two bottom rows.

Fault like shorts of adjacent address inputs generates an error in the area designated as P of different values of signals at these inputs and doesn't show an error otherwise, i.e. in the area R where these values match. In the right parts of Fig. 1, a and Fig. 1, b, two versions of the P and R areas are shown. In case of change of the version in programming of LUT pair, the P and R areas are interchanged the position. It is important as in these areas opposite results in relation to support of a checkability of the circuit and trustworthiness will be achieved. In the P area, a checkability of the circuit raises, and trustworthiness of results will be reduced. In the area R, on the contrary, the checkability of the circuit decreases, and trustworthiness of results will be increased. Thus, an opportunity to interchange the position for sets of the P and R areas allows to inscribe better them in sets of input data of the normal and emergency modes, i.e. to raise the circuit checkability in a normal mode and trustworthiness of results in emergency one.

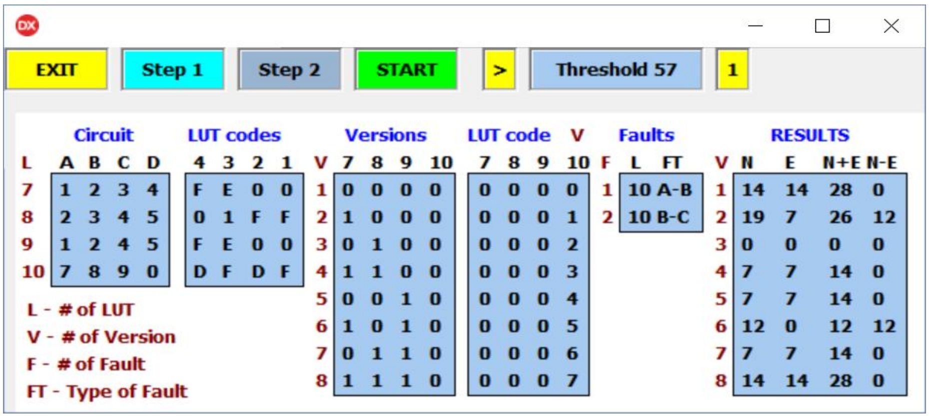
3 Program Implementation of the Offered Method

The offered method generates a set of versions in programming of LUT and provides a choice of the version which improves a digital component of safety-related system on two attributes: checkability of the circuit and trustworthiness of results.

Program implementation of this method defines all versions of a code of programming for the given hardware implementation of the LUT-oriented architecture of the FPGA project and all faults like shorts on which versions exert impact.

Further in the course of simulation of the circuit operation, the following actions are performed. For each version, the summary amount of the erroneous results received owing to action of each fault is defined. Simulation is executed for each input word. Different areas of the normal and emergency modes are researched: the amount of erratic results is calculated. Versions which as much as possible show errors in the used areas of a normal mode and exclude errors in emergency one are selected.

The demonstration option of program implementation of a method is developed using Delphi 10 Seattle demo-version [23]. Program implementation of the method is shown on the example of function evaluation in the LUT-oriented architecture supported by the CAD Altera Quartus II. Basic data for the program are the description of the circuit (amount H of inputs of the circuit, amount G of LUT and the table of connections). Inputs of the circuit and outputs of the LUT are enumerated by values 1, …, H and $H + 1$, …, $H + G$, respectively. Address inputs A, B, C, D of each LUT are connected to inputs 1, …, H of the circuit or to outputs of prior LUT units. Their numbers are written in the table of connections on intersection of number LUT and designation of its address input. In the considered example $H = 6$, $G = 4$, address inputs A, B, C, D of the LUT units 7, 8, 9 and 10 are connected to numbers 1, 2, 3, 4; 2, 3, 4, 5; 1, 2, 4, 5 and 7, 8, 9, 0, respectively. Value 0 means absence of connection. The input 6 in this part of the circuit isn't used. The panel of step by step execution of the method is shown in Fig. 2 after execution of all steps. After start of the program, its main panel shows the table of the circuit connections and the table "LUT Codes" in a hexadecimal number system.



Circuit

L	A	B	C	D
7	1	2	3	4
8	2	3	4	5
9	1	2	4	5
10	7	8	9	0

LUT codes

4	3	2	1
F	E	0	0
0	1	F	F
F	E	0	0
D	F	D	F

L - # of LUT
V - # of Version
F - # of Fault
FT - Type of Fault

Versions

V	7	8	9	10
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	1	1	0	0
5	0	0	1	0
6	1	0	1	0
7	0	1	1	0
8	1	1	1	0

LUT code V

7	8	9	10
0	0	0	0
0	0	0	1
0	0	0	2
0	0	0	3
0	0	0	4
0	0	0	5
0	0	0	6
0	0	0	7

Faults

F	L	FT
1	10	A-B
2	10	B-C

RESULTS

V	N	E	N+E	N-E
1	14	14	28	0
2	19	7	26	12
3	0	0	0	0
4	7	7	14	0
5	7	7	14	0
6	12	0	12	12
7	7	7	14	0
8	14	14	28	0

Fig. 2. Program implementation of the offered method

On a step 1 the table "Versions" in which all possible versions in programming of the circuit on FRGA are described is built. For each version 1, …, 8, the LUT, inverting the output are marked by units. Besides, the table "LUT Codes Versions" in which for LUT 2 of pairs are specified the mask codes allowing to replace value of memory bits according to need of compensating the inverse values at address inputs is built. On a step 2 the table "Faults" showing the list of faults depended on versions of programming is built. The rows of this table contain two elements: number of LUT and type of faults: A-B, B-C or C-D, i.e. shorts between address inputs of *A* and *B*, *B* and *C* or *C* and *D*, respectively. On following step "Start", simulation of the circuit operation at all input values of an operating mode for all versions of programming and all set of faults is executed. After simulation the threshold separating the normal and emergency mode is set and the table "Results" which shows quantity *N* and *E* of errors appearing respectively in normal and emergency operation and also their amount $N + E$ and a difference $N - E$ is created. Results of simulation show that version 1 contains identical quantity of errors (14) in both modes. Version 2 provides the best value of checkability (19 errors) in a normal mode. Version 6 provides the best trustworthiness (0 errors) in emergency mode with an average value of checkability (12 errors).Version 6 is preferable in case of use in normal and emergency mode of the same version of the circuit programming. Functioning of the circuit can be ensured with use of different versions in normal and emergency mode [24]. In this case it is expedient to apply versions 2 and 6 in normal and emergency mode, respectively.

4 Conclusions

The method for improving of a checkability of the circuit and trustworthiness of results in digital components of safety-related systems is offered. The digital components developed on FPGA with the LUT-oriented architecture are considered. The method allows to detect better faults of digital circuits in a normal mode and to reduce activity of faults in emergency mode of system operation. Thus, the method increases the functional safety of safety-related systems and the objects of the increased risk controlled by them. At the same time, not only the method, but also questions of its development are important. How this method was received? What objective conditions shall develop for development of this method?

The method improves two attributes which are traditionally opposed each other. The high checkability of the circuit allows to detect timely permanent faults, but promotes hyperactivity of the transient faults reducing trustworthiness of results. Overcoming negative effect of such opposition of attributes became possible in safety-related systems as a result of diversification of an operating mode by its division on normal and emergency. It led to diversification of the input data and requirements imposed to a checkability of the circuit and trustworthiness of results. Opposition of two attributes became the useful to support of the functional safety of safety-related systems.

However, this opposition shall be opposite for the normal and emergency modes where checkability and trustworthiness shall dominate respectively. Faults diversify input data by their division into two sets defining the correct and erroneous result. However this diversification doesn't correlate with diversification of input data for the normal and emergency modes. Coordination of these diversifications became possible with development of integrated circuits with a programmable logic. Version redundancy of the program code in FPGA projects has completed the formation of objective conditions for development of the offered method providing a choice of versions for domination of a checkability in a normal mode, and trustworthiness – in emergency mode.

Thus, development of offered method became possible as a result of development of resources – models, methods and means – at the level of diversification.

References

1. IEC 61508-1:2010. Functional Safety of Electrical / Electronic / Programmable Electronic Safety Related Systems – Part 1: General requirements. Geneva: International Electrotechnical Commission (2010)
2. 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)
3. Drozd, M., Drozd, A.: Safety-Related Instrumentation and Control Systems and a Problem of the Hidden Faults. In: 10th International Conference on Digital Technologies (DT'2014), pp. 137-140. Zhilina, Slovakia (2014), DOI: 10.1109/DT.2014.6868692
4. Drozd, A., Drozd, M., Antonyuk, V.: Features of hidden fault detection in pipeline digital components of safety-related systems. In: CEUR Workshop Proceedings of the 12th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, pp. 476-485, Kyiv, Ukraine (2015)
5. Sklyar, V.V., Kharchenko, V.S.: Fault-Tolerant Computer-Aided Control Systems with Multiversion-Threshold Adaptation: Adaptation Methods, Reliability Estimation, and Choice of an Architecture. In: Automation and Remote Control, vol. 63, No. 6, pp. 991-1003 (2002)
6. Drozd, J., Drozd, A., Antoshchuk, S.: Green IT engineering in the view of resource-based approach. In book: Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control, V. Kharchenko, Y. Kondratenko, J. Kacprzyk (Eds.), Vol. 74. Berlin, Heidelberg: Springer International Publishing, 43-65 (2017), DOI: 10.1007/978-3-319-44162-7_3

7. 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: 9 th International Conference on Digital Technologies (DT'2013), pp. 176-180. Zhilina, Slovakia (2013), DOI: 10.1109/DT.2013.6566307
8. Kharchenko, V., Gorbenko, A., Sklyar, V., Phillips, C.: Green Computing and Communications in Critical Application Domains: Challenges and Solutions. In: 9 th International Conference on Digital Technologies (DT'2013), pp. 191-197. Zhilina, Slovakia (2013)
9. Hahanov, V., Litvinova, E., Chumachenko, S., Liubarskyi, M.: Qubit description of the functions and structures for computing. In: 2016 IEEE East-West Design & Test Symposium (EWDTS), pp. 1-6. Yerevan, Armenia (2016), DOI: 10.1109/EWDTS.2016.7807659
10. Kondratenko, G., Kondratenko, Y., Romanov, D.: Fuzzy models for capacitative vehicle routing problems in uncertainty. In: International DAAAM Symposium, pp. 205-206 (2006)
11. Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A.: Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic. In: Kwiecien, A., Gaj, P., Stera, P. (eds.) 20th Conference on Computer Networks, CN 2013, Lwówek Slaski, Poland.Communications in Computer and Information Science, vol. 370, pp. 146-156. Springer-Verlag Berlin Heidelberg (2013), DOI: 10.1109/IDAACS.2013.6662707
12. Cyclone FPGA Family Data Sheet. Altera Corporation (2003), http://www.altera.com
13. Xilinx Staff. Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet (2005)
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. Kharchenko, V., Siora, A., Sklyar, V.: Design and Testing Technique of FPGA-based critical systems. In: 10th IEEE International Conference The Experience of Designing and Application of CAD Systems in Microelectronics, pp. 305-314. Lviv, Ukraine (2009)
16. 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)
17. Melnyk, A., Melnyk, V.: Self-Configurable FPGA-Based Computer Systems. Advances in Electrical and Computer Engineering, vol. 13, no. 2, pp. 33-38 (2013)
18. Drozd, A., Drozd, M., Kuznietsov, M.: Use of Natural LUT Redundancy to Improve Trustworthiness of FPGA Design. In: CEUR Workshop Proceedings, vol. 1614, 12th International Conference on ICT in Education, Research and Industrial Applications, ICTERI 2016, pp. 322-331, Kyiv, Ukraine (2016)
19. Pleskacz, W.A., Jenihhin, M., Raik, J., Rakowski, M., Ubar, R., Kuzmicz, W.: Hierarchical Analysis of Short Defects between Metal Lines in CMOS IC. In: 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, pp. 729-734. Parma, Italy (2008)
20. Drozd, J., Drozd, A., Antoshchuk, S., Kushnerov, A., Nikul, V.: Effectiveness of Matrix and Pipeline FPGA-Based Arithmetic Components of Safety-Related Systems. In: 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. pp. 785-789. Warsaw, Poland (2015), DOI: 10.1109/IDAACS.2015.7341410
21. Drozd, J., Drozd, A., Maevsky, D., Shapa, L.: The Levels of Target Resources Development in Computer Systems. In: IEEE East-West Design & Test Symposium, pp. 185-189. Kiev, Ukraine (2014), DOI: 10.1109/EWDTS.2014.7027104
22. Maevsky, D. A.: A New Approach to Software Reliability. Lecture Notes in Computer Science: Software Engineering for Resilient Systems, № 8166. Berlin: Springer International Publishing, 156–168 (2013), DOI: 10.1007/978-3-642-40894-6_13
23. Delphi 10 Seattle: Embarcadero (2015) https://www.embarcadero.com/ru/products/delphi
24. Drozd, O.V., Nesterenko, S.A., Drozd, J.V., Zashcholkin, K.V., Kuznetsov, M.O.: Programmable device. UA 107437, G 06 F 11/263, Patent of Ukraine. In: Bulletin № 24 (2014)