A Method of Common Signal Monitoring in FPGA-Based Components of Safety-Related Systems

Oleksandr Drozd[0000-0003-2191-6758], Viktor Antoniuk[0000-0001-8436-5338], Svetlana Antoshchuk[0000-0002-9346-145X], Myroslav Drozd[0000-0003-0770-6295]

Odessa National Polytechnic University, Ave. Shevchenko 1, 65044 Odessa, Ukraine
drozd@ukr.net, viktor.v.antoniuk@gmail.com, asgonpu@gmail.com, myroslav.drozd@opu.ua

Abstract. Traditional solutions in ensuring the functional safety of safety-related systems and their digital components based on methods and means of testing and on-line testing, as well as fault-tolerant structures, including majority schemes using multi-version technologies to counter common cause failures are considered. The limitation of these approaches by the logical checkability of digital circuits in the structural, structurally functional, and dual-mode versions is shown. Multi-version solutions are aimed at countering common cause failures, including common control faults related to reset, synchronization signals and other common signals that can block digital components and their checking circuits in a state identified as working. However, faults in chains of common signals can also be addressed to hidden faults, which remain a problem in safety-related systems. The logical checkability of the circuits decreases from structural to dual-mode and increases with the reduction of matrix structures. The maximum reduction is achieved in bitwise pipelines. The successes of green and FPGA technologies created the conditions for the development of on-line testing methods based on an assessment of energy consumption. These methods can significantly complement the logical checking. A method for monitoring common signals by estimating consumption currents in circuits of bitwise pipelines using the example of a shifting register is proposed. The results of experimental confirmation of the effectiveness of the proposed method is achieved.

Keywords: safety-related system, component, FPGA design, logical and power-oriented checkability, hidden faults, common signal, matrix structure, consumption current, bitwise pipeline

1 Introduction

One of the most important areas in the development of computer systems is associated with their critical applications, where they are used as instrumentation and control safety-related systems for managing high-risk objects that are widely represented in transport, energetics and other manufacturing and service industries [1, 2]. International standards governing the design and use of safety-related systems, impose on them the task of ensuring functional safety of the object and the system itself to prevent accidents and reduce losses in the event of their occurrence [3, 4].

Safety is provided by a set of test maintenance procedures for system components during work breaks and their continuous automatic monitoring based on on-line testing methods and tools [5, 6]. Components are designed using fault-tolerant solutions based on correction codes, reconfiguration [7, 8], as well as majority structures and multi-version technologies to counter common cause failures. Such failures include design errors and faults in chains of common signals that provide reset, synchronization, and other general control functions. Multi-version technologies are based on various types of diversity and are aimed at eliminating a common cause or the same effects in different versions of performing calculations [9].

As a rule, testing and on-line testing is performed by analyzing the result for an error, which relates the appropriate methods and means to the logical checking of digital circuits [10, 11]. The source of error is the fault of the circuit. Therefore, the logical checking is performed if the circuit fault can be observed in the calculated result, causing an error in it. This property of the suitability of the circuit to a certain type of checking of its faults is called checkability [12, 13].

Thus, the logical checking of the circuit can be performed only within its logical checkability. This conclusion also applies to the effectiveness of fault-tolerant structures and the multi-version technologies used in them [14]. A fault that does not cause an error in the analyzed result is not dangerous for computer systems working only in the operating mode, but creates the problem of hidden faults in safety-related systems designed for operation in two modes: normal and emergency. These faults, including faults in chains of common signals, can accumulate over the course of an extended normal mode and manifest themselves in emergency mode, violating the fault tolerance of the circuits and the safety of systems and control objects [15, 16].

The need to support logical checking with other forms of checking and checkability finds a response in the modern level of development of green technologies [17, 18] and FPGA design [19, 20]. At the intersection of these areas, the evaluation of circuits for their energy consumption is improving.

On-line testing methods based on measuring the temperature of the circuits in the course of their work are known [14]. We propose a method that continues this direction in relation to faults in chains of common signals based on the capabilities of modern CAD systems in estimating consumption currents.

Faults in chains of common signals have a significant impact on the energy consumption of the circuit. Monitoring of common signals is carried out within the frame of power-oriented checkability, which marks the bottom level of current consumption with proper functioning of the circuit. The proposed method detects the impossible (with the correct operation) decrease in dynamic component of the current consumption in case of a fault in the chain of common signal.

Section 2 describes logical checking problems that limit the ability to detect faults in chains of common signals for safety-related systems and highlights bitwise pipelines that show the greatest logical checkability. Section 3 notes the possibilities of modern FPGA design in the evaluation of circuits in their power consumption and formulates the main points of the method of common signals monitoring. Section 4 presents the results of experiments on the use of the suggested method in a bitwise pipeline using the example of detecting synchronization errors of a shifting register.

2 Problems of Logical Checking in Common Signal Monitoring

Logical checking problems begin with its checkability. At the same time, the restrictions imposed on the checking of the circuit by appropriate checkability are also the motivation for its increase.

Logical checkability can be refined for testing and on-line testing as forms of logical checking.

Circuit testing is limited to testability, i.e. the suitability of the circuit for developing tests that detect faults. Testability refers to logical checkability, which is characterized as structural, since it is completely determined by the structure of the circuit. Structural checkability is enhanced by methods and means of testable design, which is aimed at the formation of the structure of the scheme with a high degree of controllability and observability of its internal points [21, 22].

On-line testing is limited to logical checkability, which is characterized as structurally functional, since it is determined not only by the structure of the circuit, but also depends on its input data. Therefore, an increase in the structurally functional checkability of the circuit can be achieved on the basis of two approaches: the improvement of its structure, and a change in the character of the input data.

The matrix structure orients the digital circuit to receive and process of input data in parallel codes. This introduces major limitations to the controllability and observability of the interior points of the circuit. Therefore, the first approach, as a rule, is based on the reduction of matrix structures. The processing of approximate data in floating-point formats [23, 24] demonstrates this approach in the use of truncated arithmetic operations with mantissas [25]. Residue checking of abbreviated operations performed in matrix arithmetic nodes is described in [26, 27].

Both approaches are implemented in bitwise pipelines, which reduce the matrix of operational elements in the pipeline section to one element. At the same time, the nature of the input data changes, which are represented by sequential codes. The structure of bitwise pipelines is based on scanning registers, which are elements of testable design [28].

The fault-tolerant circuits used in the digital components of safety-related systems are also limited by logical checkability, which reflects the characteristics of these systems to separate the operating mode into normal and emergency [29].

The result of this separation is the following chain of consequences:

- Various input circuit data in these modes;
- Various structurally functional checkability, characterized as dual-mode;
- A new type of faults hidden in normal mode due to the lack of input data showing them and violating fault tolerance of circuits on other input data in emergency mode.

The problem of hidden faults has not yet received a safe solution. Detection of hidden faults that can accumulate in the chains of common signals is still performed using simulation modes, that recreate emergency conditions and not once led to them as a result of unauthorized activation by malfunction or with human participation [30, 31].

It should be noted that testability, as the simplest form of logical checkability, is the greatest. Structurally functional checkability is limited from above by testability

because it has an additional limiting dependence on the input data. Dual-mode checkability is the least. It is bounded above by the structurally functional checkability of the normal mode, since does not consider faults hidden in emergency mode.

Therefore, it is important to increase dual-mode checkability both at the level of testability and at the level of structurally functional checkability. This solution is to perform calculations on bitwise pipelines:

- Their register structures combine scanning register functions that increase testability in testable design;
- Minimal matrix structures in the sections of the pipeline increase the structurally functional checkability of the circuit;
- Processing data in sequential codes evens out the variety of input data and structurally functional checkability of the circuit in normal and emergency modes and in this way improves dual-mode checkability.

Thus, logical checking is not sufficiently ensured by the checkability of circuits in the components of safety-related systems and needs to be supplemented with other forms of checking, in particular, to detect hidden faults in chains of common signals. It should be noted that from the point of view of logical checking, the most checkable solution for safety-related systems is the design of their digital components based on bitwise pipelines.

3 Monitoring of the Common Signals

The method of monitoring of the common signals is based on power-oriented checkability, which is provided by the circuit with the support of FPGA design methods and tools developed in green technologies. Modern FPGA-oriented CAD systems contain utilities that model a FPGA project to estimate its power consumption. The simulation results are estimates of currents consumption: the total current and its dynamic and static components. These results can be obtained for different activity of the input signals, which is given as a percentage of the activity of the clock signals. Reducing the activity of the input signals leads to a decrease in current consumption.

Fault detection in the chain of common signals is based on the manifestation of this fault in the form of a decrease in current consumption in its dynamic component due to a decrease in the number of switching signals in the circuit. Such a manifestation of a fault is characteristically for synchronization signals, reset signals and other general control signals. Fault of the common signal blocks the operation of the circuit or its part, as a result of which we can expect a significant decrease in switching activity, as well as the dynamic component and the total current consumption.

The method detects a fault if the current consumption drops below the minimum possible level when properly functioning. This threshold $I_{PT.MIN}$ of fault detection is marked by power-oriented checkability, which takes into account:

- The minimum total current consumption I_{PT}, which is determined by simulation at zero activity of input signals;
- The error δI_{PT} of simulation when estimating total current consumption;
- The error δI_M of measuring the current consumption by the sensor in the process of common signals monitoring.

The fault detection threshold is determined by the following formula:

$$I_{PT.MIN} = I_{PT} - \delta I_{PT} - \delta I_M.$$

As a rule, CAD systems simulate and sensors measure the current consumption with an error not exceeding $\delta I_{PT} = \delta I_M = 0.5\%$, which determines the fault detection threshold at the level: $I_{PT.MIN} = 0.99\ I_{PT}$.

We consider the proposed method for a bitwise pipeline on the example of a shifting register to the right, which consists of n series-connected triggers, numbered from 1 to n from left to right. The dynamic component I_{PTD} of current consumption consists of two parts I_{DC} and I_{DI}, which are determined by switching the clock signals and information signals, respectively: $I_{PTD} = I_{DC} + I_{DI}$. The switching of information signals is taken into account when setting the activity of the input signals as a certain number of z as a percentage of the activity of the clock signals: $I_{DI} = z\ I_{DC}$.

Then the dynamic component is represented as $I_{PTD} = I_{DC}\ (1 + z)$.

Let the fault turn off the clock signals for d triggers. Then we can expect the decrease of the current I_{DC} by value $\Delta I_{DC} = I_{DC}\ r$, where $r = d / n$. I_{DI} current decreases by value $\Delta I_{DI} = I_{DI}\ (n - x + 1) / n$, where x is the smallest number within the disabled triggers numbers, $x \le n - d + 1$, since all triggers starting from the disabled x trigger will have zero information activity signals.

The decrease ΔI_{PT} of the total current consumption, which is manifested in the reduction of the dynamic component ΔI_{PTD} while maintaining the static component, is determined by the following formula:

$$\Delta I_{PT} = \Delta I_{PTD} = I_{DC}\ (d + z\ (n - x + 1)) / n. \quad (1)$$

The possibilities of the proposed method are governed by the minimum decrease $\Delta I_{PTD\ MIN}$ in the dynamic component of the current consumption, which is achieved under the condition $x = n - d + 1$, i.e. in the case of the location of all disabled triggers in a row at the right end of the shifting register.

The minimum decrease of the total current consumption and its dynamic component is determined from the formula (1) as follows:

$$\Delta I_{PT\ MIN} = I_{DC}\ (z + 1)\ d / n = I_{PTD}\ r. \quad (2)$$

Thus, we can expect a decrease in total current consumption by at least a fraction of the dynamic component, equal to the fraction r of the disabled triggers. We can estimate the values of r and $d = r\ n$, based on the inequality $I_T - I_{PT.MIN} < \Delta I_{PT\ MIN}$, where I_T – is the total current consumption resulting from the simulation. Then, according to the formula (2), $r > (I_T - I_{PT.MIN}) / I_{PTD}$ and

$$d > n\ (I_T - I_{PT.MIN}) / I_{PTD}.$$

The method uses the total current consumption I_M, measured by the sensor [32], and the fault detection threshold $I_{PT.MIN}$. In case of $I_M < I_{PT.MIN}$, the method identifies the fault.

4 Experimental Assessment of Suggested Method

For the experimental evaluation of the proposed method in the Quartus Prime 18.1 Lite Edition CAD system [33] the VHDL-project of a 32-bit shifting register with information input *SI*, reset input *Res*, clock inputs *CLK* for each individual register trigger, and information *SO* output has been created (Fig. 1).

The project was implemented on Intel Max 10 FPGA 10M50DAF672I7G [34]. Setting the time characteristics was made in the utility TimeQuest Timing Analyzer [35].

Setting activity values A_I of the input and internal signals of the circuit, and modeling the total current consumption I_T of FPGA core, as well as its dynamic I_D and static I_S components, was performed by the PowerPlay Power Analyzer utility [36].

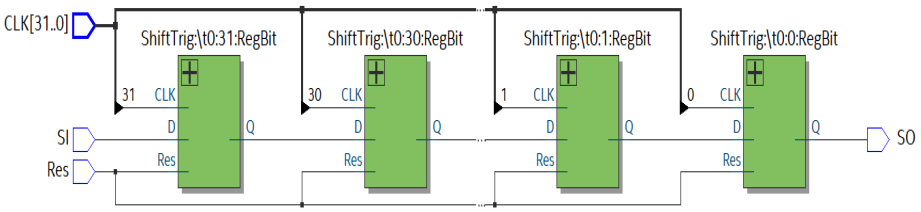

Fig. 1. An example of the RTL-scheme of a 32-bit shifting register

The simulation consisted in determining the values of the current consumption of the circuit of shifting register for all properly functioning 32 bit ($d = 0$) and then disabling the CLK inputs of triggers at the right end of the register starting with row of $d = 4$ triggers with a shutdown step of 4. The values of currents consumption for all cases determines with range of activity A_I of the input information signal *SI* and internal shifting register signals from 0% to 100% of the clock signal frequency *CLK*, which was 250 MHz.

Currents I_T, I_S, I_D modelling results are given in Table 1.

Tab. 1. Experiment results for 32-bit shifting register

A_I, %	$d = 0$			$d = 4$			$d = 8$			$d = 12$		
	I_D, mA	I_S, mA	I_T, mA	I_D, mA	I_S, mA	I_T, mA	I_D, mA	I_S, mA	I_T, mA	I_D, mA	I_S, mA	I_T, mA
0	5.77	11.81	**17.59**	5.22	11.80	**17.02**	4.23	11.79	**16.02**	3.63	11.78	**15.41**
12.5	6.48	11.82	18.30	5.75	11.80	17.55	4.68	11.79	**16.47**	4.00	11.78	**15.78**
25	7.18	11.82	19.01	6.28	11.81	18.08	5.13	11.79	**16.93**	4.36	11.78	**16.14**
37.5	7.89	11.83	19.71	6.80	11.81	18.61	5.58	11.79	**17.38**	4.73	11.78	**16.51**
50	8.59	11.83	20.42	7.33	11.81	19.14	6.04	11.80	17.83	5.09	11.78	**16.88**
62.5	9.30	11.84	21.13	7.86	11.81	19.67	6.49	11.80	18.29	5.46	11.79	**17.24**
75	10.00	11.84	21.84	8.38	11.81	20.20	6.94	11.80	18.74	5.82	11.79	17.61
87.5	10.70	11.85	22.55	8.91	11.82	20.73	7.39	11.80	19.19	6.19	11.79	17.98
100	11.41	11.85	23.26	9.44	11.82	21.26	7.84	11.80	19.65	6.55	11.79	18.34

In Table 1, value $I_{PT} = 17.59$ of current I_T for the case $d = 0$ and $A_I = 0$ is highlighted in bold. This value is used to calculate the fault detection threshold $I_{PT.MIN} = 17.41$. In the columns, where $d > 0$, the values of the current I_T is highlighted in bold when it is below the $I_{PT.MIN}$ threshold, i.e. in case of fault detection.

Table 2 compares the calculated and experimental results.

Tab. 2. Comparison of the calculated and experimental results

A_I, %	I_D, mA	I_S, mA	$I_{T\ d=0}$, mA	$I_{T\ d=4}$, mA	$I_{T\ d=8}$, mA	$I_{T\ d=12}$, mA	$\delta_{T\ d=4}$, %	$\delta_{T\ d=8}$, %	$\delta_{T\ d=12}$, %
0	5.77	11.81	17.59	16.87	16.15	15.43	0.88	-0.81	-0.13
25	7.18	11.82	19.01	18.11	17.21	16.32	0.17	-1.65	-1.12
50	8.59	11.83	20.42	19.35	18.23	17.20	-1.10	-2.47	-1.90
75	10.00	11.84	21.84	20.59	19.34	18.09	-1.93	-3.20	-2.73
100	11.41	11.85	23.26	23.26	20.41	18.98	-2.68	-3.87	-2.44

The total current consumption $I_{T\ d=4,\ 8,\ 12}$ when d triggers are disconnected is calculated by reducing the current $I_{T\ d=0}$ by the value $\Delta I_{PT\ MIN} = I_D\, d\ /\ 32$. This expected current value is compared with the current I_T, obtained for the corresponding d experimentally (Table 1). The error $\delta_{T\ d=4,\ 8,\ 12}$ is calculated by the formula:

$$\delta_{T\ d} = (I_{T\ d} - I_T)\ /\ I_T.$$

Fig. 2 visualize the values of calculated and experimental I_T for $d = 8$. Fig 3 visualize the values of error δ_T for $d = 4, 8, 12$.

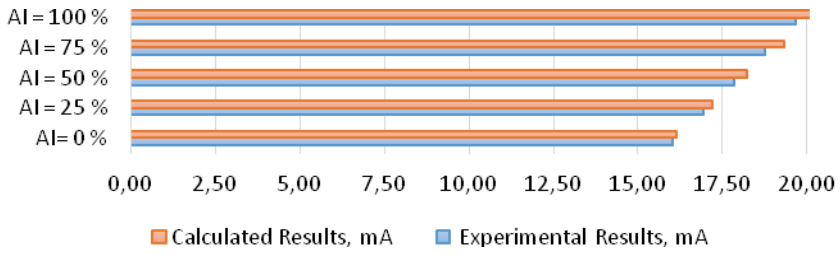

Fig. 2. Visualization of the values I_T for $d = 8$

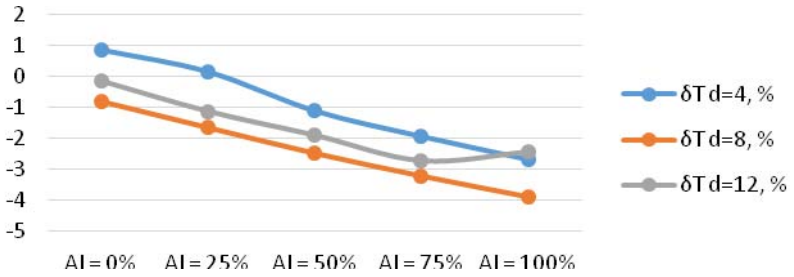

Fig. 3. Visualization of the calculated and experimental results for various *d*

The obtained error values indicate a good agreement between the theoretical and experimental results that characterize the suggested method for monitoring common signals.

5 Conclusions

Methods and tools for testing and on-line testing, as well as fault-tolerant solutions are the basis for ensuring the functional safety of safety-related systems. However, the possibilities of these approaches, combined by the features of logical checking, are significantly limited by the structural, structurally functional, and dual-mode checkability of digital circuits, respectively.

The lack of dual-mode checkability, the carriers of which are matrix digital circuits in the components of safety-related systems, creates in them the problem of hidden faults. These faults pose a significant threat to the safety of systems and control objects in critical applications. The use of hazardous imitation modes that recreate emergency conditions to detect hidden faults is an indisputable proof of the significance of this problem, which prevents fault-tolerant solutions from becoming fault-safe.

In conditions of limited structurally functional checkability of digital circuits in normal mode, many faults can become hidden. In this case, faults in chains of common signals can cause serious functional disturbances in the system components and manifest themselves with the transition to emergency mode. Means of online testing can be blocked by such faults or taken for the correct state of the circuit, fixed by the fault.

Logical checking of circuits in critical applications requires the involvement of other forms of checking.

The suggested method of common signals monitoring continues the development of approaches that take into account the energy impact of faults on the operation of the circuit. Known solutions that perform temperature monitoring of circuits take into account changes in power dissipation using temperature sensors. The proposed method uses the achievements of green and FPGA technologies, which allow to detect the energy trace of faults based on much more accurate estimates of current consumption.

The method is shown for bitwise pipelines that have the greatest logical checkability in critical applications, using the example of a shifting register and a fault in the synchronization of its triggers. The results of the experiments have showed high efficiency of the method and their good convergence with the theoretical estimates of the method.

References

1. Gorbenko, A., Kharchenko, V., Tarasyuk, O., Zasukha, S.: Safety of Rocket-Space Engineering and Reliability of Computer Control Systems and Software: 2000-2009 YRS. In: First Intern. Workshop "Critical Infrastructure Safety and Security", pp. 79–93, Kirovograd, Ukraine (2011).
2. Efanov, D., Lykov, A., Osadchy, G.: Testing of relay-contact circuits of railway signalling and interlocking. In: IEEE East-West Design and Test Symposium, EWDTS 2017, pp. 242–248, Novi Sad, Serbia (2017).
3. International Electrotechnical Commission, Nuclear Power Plants: Instrumentation and Control for Systems Important to Safety – General Requirements for Systems, Rep. IEC 61513, IEC, Geneva (2001).
4. IAEA NS-G-1.3. Instrumentation and control systems important to safety nuclear power plants. Safety guide, Vienna (2002).
5. Romankevich, V.: Self-testing of multiprocessor systems with regular diagnostic connections. Automation and Remote Control, vol. 78, no. 2, pp. 289–299 (2017).
6. Abramovichi, M., Stroud, C., Hamiliton, C., Wijesuriya, S., Verma, V.: Using roving STARs for on-line testing and diagnosis of FPGAs in fault-tolerant applications. In: IEEE International Test Conference, pp. 973–982. Atlantic City, USA (1999).
7. Yatskiv, V., Yatskiv, N., Jun, S., Sachenko, A., Zhengbing, H.: The use of modified correction code based on residue number system in WSN. In: 7th IEEE International Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 513–516. Berlin, Germany (2013).
8. Palagin, A.V., Opanasenko V.N.: Design and application of the PLD-based reconfigurable devices. In: Adamski, M., Barkalov, A., Wegrzyn M. (eds.) Design of Digital Systems and Devices, Lecture Notes in Electrical Engineering, vol. 79, pp. 59–91. Springer Verlag, Berlin Heidelberg (2011).
9. Kharchenko, V.: Multy-version Systems: Models, Reliability, Design Technologies. In: 10th European Conference on Safety and Reliability, pp. 73–77, Munich, Germany (1999).
10. Brezhnev, E.: An approach for assessing risk of common cause failures in critical infrastructure. Information & Security, vol. 28, no. 1, 199–210 (2012).
11. Hahanov, V., Litvinova, E., Obrizan, V., Gharibi, W: Embedded method of SoC diagnosis. Elektronika in Elektrotechn, no 8, pp. 3–8 (2008).
12. Drozd, A., Drozd, M., Martynyuk, O., Kuznietsov, M.: Improving of a Circuit Checkability and Trustworthiness of Data Processing Results in LUT-based FPGA Components of Safety-Related Systems. In: CEUR Workshop Proceedings, vol. 1844, pp. 654–661 (2017).
13. Drozd, A., Antoshchuk, S., Drozd, J., Zashcholkin, K., Drozd, M., Kuznietsov, N., Al-Dhabi, M., Nikul, V.: Checkable FPGA Design: Energy Consumption, Throughput and Trustworthiness. In book: Green IT Engineering: Social, Business and Industrial Applications, Studies in Systems, Decision and Control, V. Kharchenko, Y. Kondratenko, J. Kacprzyk (edits), vol. 171, pp. 73–94. Berlin, Heidelberg: Springer International Publishing (2018). DOI: 10.1007/978-3-030-00253-4_4.
14. Nicolaidis, M., Zorian, Y., Pradhan, D (eds): On-Line Testing for VLSI. In: Journal of Electronic Testing: Theory and Application, vol. 12, no. 1/2, pp. 7–159 (1998).

15. Drozd, A., Drozd, M., Antonyuk, V.: Features of Hidden Fault Detection in Pipeline Components of Safety-Related System, CEUR Workshop Proceedings, vol. 1356, pp. 476–485 (2015).
16. Drozd, A., Drozd, J., Antoshchuk, S., Antonyuk, V., Zashcholkin, K., Drozd, M., Titomir, O.: Green Experiments with FPGA. In book: Green IT Engineering: Components, Networks and Systems Implementation, V. Kharchenko, Y. Kondratenko, J. Kacprzyk (eds), vol. 105, pp. 219–239. Berlin, Heidelberg: Springer International Publishing (2017). DOI: 10.1007/978-3-319-55595-9_11.
17. Murugesan, S., Gangadharan, G.: Harnessing Green IT. Principles and Practices, UK: Wiley and Sons Ltd. (2012).
18. Tyurin, S., Kamenskih, A.: Green Logic: Models, Methods, Algorithms. In: Kharchenko V., Kondratenko Y., Kacprzyk J. (eds) Green IT Engineering: Concepts, Models, Complex Systems Architectures. Studies in Systems, Decision and Control, vol 74, pp 69–86, Springer (2017). DOI: https://doi.org/10.1007/978-3-319-44162-7_4.
19. Drozd, A., Drozd, M., Kuznietsov, M.: Use of Natural LUT Redundancy to Improve Trustworthiness of FPGA Design. In: CEUR Workshop Proceedings, vol. 1614, pp. 322–331 (2016).
20. Kondratenko, Y., Gordienko, E.: Implementation of the neural networks for adaptive control system on FPGA. In: Annals of DAAAM for 2012 & Proceeding of the 23th Int. DAAAM Symposium "Intelligent Manufacturing and Automation", vol. 23, no.1, B. Katalinic (Ed.), DAAAM International, pp. 389–392, Vienna, Austria (2012).
21. Matrosova, A., Nikolaeva, E., Kudin, D., Singh, V.: PDF testability of the circuits derived by special covering ROBDDs with gates. In: IEEE East-West Design and Test Symposium, EWDTS, pp. 1–5. Rostov-on-Don, Russia (2013).
22. Goldstein, L. M. Thigen, E. L.: SCOAP: Sandia Controllability/Observability Analysis Program. In: 17th Design Automation Conference, pp. 190–196, Minneapolis, MN, USA, (1980).
23. ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic (1985).
24. IEEE Std 754™-2008 IEEE Standard for Floating-Point Arithmetic. IEEE 3 Park Avenue New York, NY 10016–5997, USA (2008).
25. Garofalo, V.: Truncated Binary Multipliers with Minimum Mean Square Error: Analytical Characterization, Circuit Implementation and Applications. In: Ph.D. Dissertation. University of Studies of Naples “Federico II”, Naples, Italy (2008).
26. Drozd, A., Lobachev, M., Hassonah, W.: Hardware Check of Arithmetic Devices with Abridged Execution of Operations. In: European Design and Test Conf, p. 611. Paris, France (1996) DOI: 10.1109/EDTC.1996.494375.
27. Drozd, A., Lobachev, M.: Efficient On-line Testing Method for Floating-Point Adder. In: Design, Automation and Test in Europe. Conference and Exhibition 2001, pp. 307–311. Munich, Germany (2001). DOI: 10.1109/DATE.2001.915042.
28. Bennets, R. G., Maunder, C. M., Robinson, G. D.: CAMELOT: a computer-aided measure for logic testability. In: IEE Proceedings E – Computers and Digital Techniques, vol. 128, no 5, pp. 177–189 (1981).
29. Kharchenko, V., Gorbenko, A., Sklyar, V., Phillips, C.: Green Computing and Communications in Critical Application Domains: Challenges and Solutions. In: 9th International Conference on Digital Technologies (DT’2013), pp. 191–197. Zhilina, Slovakia (2013).
30. Gillis, D.: The Apocalypses that Might Have Been. [Online]. Available: http://www .popmech.ru/go.php?url=http%3A%2F%2F www.damninteresting.com%2F%3Fp%3D913.
31. U.S.-Canada Power System Outage Task Force: Final Report on the August, 14, 2003 Blackout in the United States and Canada: Causes and Recommendations USA, (2004).
32. MAX 10 FPGA Development Kit User Guide (2017), https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-max10m50-fpga-dev-kit.pdf, last accessed 2019/03/20.

33. Intel Quartus Prime Standard Edition User Guide: Getting Started, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qps-getting-started.pdf, last accessed 2019/03/20.
34. Max 10 FPGA Device Architecture (2017), https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/max-10/m10_architecture.pdf, last accessed 2019/03/20.
35. Intel Quartus Prime Standard Edition User Guide: Timing Analyzer (2018), https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qps-timing-analyzer.pdf, last accessed 2019/03/20.
36. Intel Quartus Prime Standard Edition User Guide: Power Analysis and Optimization (2018), https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qps-power.pdf, last accessed 2019/03/20.