Power-Oriented Checkability of Matrix and Pipeline Circuits in FPGA-Based Digital Components of Safety-Related Systems

Oleksandr Drozd, Viktor Antoniuk, Miroslav Drozd, Hanna Stepova

Institute of Computer Systems, Odessa National Polytechnic University,
Avenue Shevchenko 1, 65044 Odessa, Ukraine
drozd@ukr.net, viktor.v.antoniuk@gmail.com, miroslav_dr@mail.ru,
hanna.suhak@gmail.com

Abstract. The checkability of the circuits is considered as a necessary condition for ensuring functional safety for safety-related systems based on the use of fault-tolerant solutions. The features of logical checkability, which is essential for testing, testable design and on-line testing of digital components of safety-related systems, are analyzed. Logical checkability is represented as structural, structurally functional and dual-mode, typical for critical applications. The problem of hidden faults is noted, which shows the lack of dual-mode checkability in the design of digital components based on matrix structures. The resource-based approach identifies this problem as a growth problem, the solution of which requires the reduction of matrix structures. The maximum reduction is achieved in bitwise pipelines. The limitations of logical checkability are shown in solving the problem of hidden faults under the conditions of the dominance of matrix structures and in the monitoring of faults in chains of the common signals. The success of green technologies in FPGA design created the conditions for the development of power-oriented checkability, which significantly complements the logical checkability of the circuits. An analytical evaluation of power-oriented checkability was obtained. The results of power-oriented checkability evaluation experiments are shown to be important for faults in chains of the common signals. Experiments were carried out for matrix and bitwise pipeline circuits using the example of multipliers of numbers. A comparative analysis of the results obtained.

Keywords: safety-related system, digital component, FPGA, logical and power-oriented checkability, problem of the hidden faults, resource-based approach, matrix structure, bitwise pipeline, faults in chains of the common signals

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

1 Introduction

Instrumentation and control safety-related systems are an important part of high-risk objects, which are widely represented in the energy sector, on high-speed ground and air transport by power plants, power grids, vehicles and their infrastructures. These

systems are aimed at ensuring functional safety of both the system and the control object. They play an important role in preventing accidents at high-risk objects and in reducing losses in the event of an accident [1, 2].

Functional safety is based on the use of fault-tolerant solutions [3, 4].

However, a fault-tolerant solution is not yet fault-safe. Indeed, a fault-tolerant solution is resistant to the number of failures specified in the design. If the number of failures exceeds the established threshold, then the fault-tolerant solution is no longer fault-safe. Thus, on the way from fault tolerance to functional safety is another important characteristic of the component of safety-related system. This characteristic is checkability, i.e. the suitability of the component's circuit for its faults being checked [5, 6].

Safety-related systems are distinguished from the general number of computer systems by dividing the operating mode into normal and emergency. Modern technologies used in safety-related systems are aimed at maintaining the system and the object in a normal mode throughout the entire operation time. Therefore, the most critical emergency mode is rarely activated and is poorly understood. The main question that is posed throughout the normal mode is whether the safety-related system is ready to perform its basic functions, i.e. emergency mode functions to prevent accidents at the control object and mitigate the consequences of the accident [7, 8].

Setting of this question has the complete reasons which are based on a problem of the hidden faults. This problem is the accumulation of hidden faults over the course of an extended normal mode in the absence of input data, which may manifest these faults as an error of the calculated result. With the beginning of the emergency mode, the input data changes its character and manifests accumulated faults that reduce the fault tolerance of the system components and its functional safety. Faults occurring in emergency mode, i.e. abnormal faults do not contribute to countering the accident [9].

The absence of conditions for the manifestation of faults is explained by the lack of checkability of the circuit. Checkability is defined in relation to a particular type of checking. The most widely used logical checking, which identifies a fault by its manifestation in the form of an error of the analyzed result. Logical checking is performed within the frame of its corresponding logical checkability, which can be structural, structurally functional, and dual-mode structurally functional [10, 11].

Structural checkability, which is determined only by the structure of the scheme, is testability, i.e. ability of the circuit to being tested in the pauses of its work. Known testable design is aimed at improving the structural checkability of circuits [12, 13].

In the process of performing operations, the circuit is characterized by structurally functional checkability, which depends on the structure of the circuit and on the input data. Structurally functional checkability creates the conditions for error detection by methods and means of on-line testing [14, 15].

Dual-mode structurally functional checkability is inherent to safety-related systems and is a consequence of the division of the operating mode into normal and emergency. This leads to different structurally functional checkability of digital circuits in normal and emergency mode due to the different input data received in these modes. Dual-mode structurally functional checkability consists in an ability of the circuit to show the abnormal faults in a normal mode. Dual-mode structurally functional checkability is part of the structurally functional checkability of the circuit in the normal

mode, because it does not take into account faults that manifest themselves only in the normal mode and have no consequences in the emergency mode.

Dual-mode checkability is maximum with minimal difference of structurally functional checkability of the circuit in normal and emergency mode. This difference, supported by the various input data of the circuit in these modes, serves as a source for the problem of hidden faults [11].

This problem is better known for unsuccessful attempts to detect hidden faults by using imitation modes, i.e. recreation of the accident conditions to test the operation of the safety-related system and its components in emergency mode. Unauthorized activation of imitation modes by a person or malfunction more than once led to accident consequences [16, 17].

The presence of dangerous imitation modes in the arsenal of methods for solving the problem of hidden faults indicates a lack of confidence in the fault tolerance of the components used in safety-related system and is explained by the lack of checkability of the designed circuits.

Low structurally functional checkability of the circuit is due to its structural redundancy, which is a consequence of two main reasons: the need to use fault-tolerant solutions with significant structural redundancy; limited input data in normal mode. The first reason is objective, since fault-tolerant solutions are the basis for ensuring the functional safety of safety-related systems.

However, the problem of hidden faults is related to the second reason, which only seems to be objective. Indeed, it is objective only within the frame of traditional design of digital components based on matrix structures using parallel adders and comparators, iterative array multipliers and dividers [18].

These nodes process data in parallel codes on matrices of homogeneous operational elements. Parallel codes help distinguish the input data of the circuit in normal and emergency mode. The second reason can be eliminated by reducing the matrix structures based on the gain of the pipeline parallelism of circuit solutions. Modern digital components are built pipeline, but the pipeline sections are matrix nodes or their elements [19, 20].

The reduction of the matrix structure to one operational element in the pipeline section converts the digital component into a bitwise pipeline that performs operations in sequential codes. Register structures of the bitwise pipeline, which are elements of testable design (scanning registers), significantly increase the structurally functional checkability of the circuit in normal mode. Sequential codes align the variety of input data of the normal and emergency modes, significantly increasing the dual-mode structurally functional checkability of the circuits in safety-related systems.

At the same time, logical checkability has a number of limitations associated with the continued dominance of matrix structures in the design of digital components, as well as the problems of on-line testing.

These restrictions stimulate the search for other forms of checking and checkability of components used in safety-related systems.

It should be noted the successful development of digital components of safety-related systems based on the component approach [21, 22] and orientation to FPGA design [23, 24].

The sustainable development of green information technologies in power saving [25, 26] has created the prerequisites for the implementation of checking of the cir-

cuits within the frame of checkability in power consumption. Modern CAD systems have received tools for estimating the power consumption of FPGA projects.

Therefore, a number of issues arise related to the limited of logical checkability and expediency of developing the power-oriented checkability of the circuits, as well as perspectivity of digital components based on matrix structures and bitwise pipelines.

The purpose of this study is to evaluate the expediency of developing the power-oriented checkability of the circuits under the conditions of a developed logical checkability in relation to traditional matrix and perspective bitwise pipeline circuit solutions.

The second section analyzes the limitations in the logical checkability of circuits that justify the development of power-oriented checkability. The third section defines the analytical assessment of power-oriented checkability, taking into account the possibilities offered by modern CAD systems using the example of Quartus Prime [27]. The fourth section describes the results of experiments and their comparative analysis in assessment of power-oriented checkability for FPGA projects with matrix and bitwise pipeline circuits using the example of multipliers of numbers with different size.

2 Expediency of Power-Oriented Checkability Development

Logical checking has received a monopoly both in relation to testing digital circuits, and in the domain of their on-line testing. The long-term sustainable development of logical checking has created a powerful infrastructure of models, methods and tools that support further dominance. Under these conditions, the development of alternative forms of checking and appropriate checkability becomes justified and successful in the case of obtaining its own place, where logical checking and logical checkability are limited in efficiency, and the alternative approach demonstrates the desired positive effect.

The first condition for obtaining your own place requires evaluating the logical checking from the position of its lacks that are essential for critical applications.

Among the main challenges to logical checking, the problem of hidden faults is dominant. According to the resource-based approach [28], this problem is related to growth problems. The resource-based approach considers models, methods and means as resources and identifies three levels of their development: replication, diversification and autonomy.

Replication is the lowest level of resource development. Matrix structures are stamped from homogeneous elements at the replication level. For example, the iterative array multiplier of n-bit binary numbers consists of n^2 operational elements, i.e. contains 10^3 operational elements for $n = 32$ [29, 30].

Problems of functional safety can be solved, starting with the level of diversification. Therefore, computer systems in critical applications diversify the working mode, dividing it into normal and emergency, i.e. rise to the level of diversification. However, digital components continue to be stamped at the replication level based on matrix structures. This discrepancy in the level of development of the system and components leads to the problem of hidden faults.

The solution to the problem is to develop components to the level of the system, for example, by transforming matrix circuits into bitwise pipelines, reflecting the level of diversification.

However, the development of matrix structures is protected by a powerful infrastructure that has been created for decades, combining the best solutions in this area in the form of models, methods and tools, including CAD, focused on designing matrix circuits, extensive libraries of ready-made matrix nodes, tools of accelerated addition of parallel codes and iterative array multipliers built into FPGA chips [31, 32].

Matrix infrastructure significantly limits the efficiency of bitwise pipelines designed within this framework. Therefore, it is advisable to combine the development of bitwise pipelines with the reduction of matrix structures within the frame of the existing infrastructure. This solution is the use of truncated arithmetic operations in the processing of approximate data [33, 34].

An important feature of computer systems is their dominant development along the path of processing approximate data in floating-point formats [35, 36]. Safety-related systems are such computer systems that receive raw data from sensors. Measurement results related to approximate data are also the source data for critical domains of cyber-physical systems and Internet of Things systems [37, 38].

Truncated arithmetic operations almost twice simplify matrix structures. However, reducing computations complicates on-line testing, which, as a rule, are performed by residue checking [39, 40].

We can observe a tendency towards complication of objects of logical checking and, accordingly, a decrease in the logical checkability of matrix circuits and an increase in the complexity of methods and means for on-line testing.

The following lack of logical checking is associated with faults in the chains of common signals, for example, reset or clock signals. Such faults can fix the digital component circuit in a state that is identified by the logical control as correct. Logical checkability is not sufficient to detect such faults.

Power-oriented checkability, on the contrary, is sensitive to such faults, that reducing the number of switching signals and, accordingly, the dynamic component of power consumption.

In addition, power-oriented checkability increases in contrast to logical checkability with the complexity of the circuit and the corresponding increase in power consumption. This effect in floating-point matrix circuits is supported by the quadratic dependence of the circuit complexity on the range of the data being processed. Bitwise pipelines achieve this effect at an increased frequency.

It should also be noted that power-oriented checkability is not tied to digital circuits, as is the case for logical checkability, and can serve hybrid circuits that also contain analog nodes.

Thus, power-oriented checkability significantly complements logical checkability.

At the same time, power-oriented checkability receives significant support from FPGA design systems. These systems offer intelligent power assessment tools for FPGA projects. Such support from CAD is being improved as part of the successful development of green technologies [41, 42].

3 Analytical Assessment of the Power-Oriented Checkability

The checkability of the scheme can be estimated by the ratio of the set of impossible values of the checked indicator, i.e. values that can only be obtained under the action of a fault, to the total number of values. In the case of power-oriented checkability of the circuit, the checked indicator is the power consumption, which, taking into account the constant supply voltage, is fully characterized by the current consumption and therefore will be assessed by the current consumption. The sets of impossible and all values of the consumed current are represented by the volumes of the ranges of their change from the lowest to the highest value.

The existence of two ranges of impossible values that are below and above the allowable values of current consumption determines, respectively, the lower and upper power-oriented checkability of the circuit.

We consider lower power-oriented checkability C_{LPC}, which provides monitoring of common signals, such as reset or clock, and general control. Faults in chains of the common signals can significantly reduce power consumption in its dynamic component and are not always succumb to logical checking. The C_{LPC} checkability can be estimated taking into account the smallest $I_{D\,MIN}$ and largest $I_{D\,MAX}$ possible value of the dynamic component by the following formula:

$$C_{LPC} = I_{D\,MIN} / I_{D\,MAX}. \quad (1)$$

It should be noted that the sensors measure the total current consumption $I_{T.S}$ and do not determine its dynamic component [43].

In Quartus Prime, a CAD system for designing digital circuits on Intel FPGA PLD, the current consumption of the project is estimated by the Power-Play Power Analyzer utility [44]. This utility estimates the total current consumption I_T of the PLD core and its dynamic I_D and static I_S components with an error ΔI_T, ΔI_D and ΔI_S at the level of 5%. In the process of measurement, the dynamic component can be estimated by the formula: $I_{D.S} = I_{T.S} - I_S \pm \Delta I_{T.S} / 2 \pm \Delta I_S / 2$, where $\Delta I_{T.S}$ – current consumption measurement error, I_S and ΔI_S – static component and its error determined previously by the utility Power-Play Power Analyzer. In case of proper functioning of the circuit $I_{T.S} = I_T$ and $I_{T.S} - I_S = I_D$. In addition, as a rule $\Delta I_{T.S} \le \Delta I_T$, i.e. we can accept $\Delta I_{T.S} = \Delta I_T$.

The Power-Play Power Analyzer utility estimates the consumption currents depending on the specified activity of the input signals, increasing the values of the I_T and I_D currents with increasing activity. It can be assumed that the $I_{D\,MIN}$ and $I_{D\,MAX}$ currents are achieved with zero and maximum activity of the input signals, respectively, i.e.

$$I_{D\,MIN} = I_{D.MIN} - (\Delta I_{T.MIN} + \Delta I_{S.MIN}) / 2; \quad (2)$$
$$I_{D\,MAX} = I_{D.MAX} + (\Delta I_{T.MAX} + \Delta I_{S.MAX}) / 2, \quad (3)$$

where the indices "*.MIN*" and "*.MAX*" mean currents and their errors at zero and maximum activity of the input signals, respectively.

Thus, the evaluation of the power consumption parameters of the project, performed by modeling in the Power-Play Power Analyzer utility, determines the lower power-oriented checkability of the circuit using the formulas (1) – (3) as follows:

$$C_{LPC} = (I_{D.MIN} - (\Delta I_{T.MIN} + \Delta I_{S.MIN}) / 2) / (I_{D.MAX} + (\Delta I_{T.MAX} + \Delta I_{S.MAX}) / 2). \quad (4)$$

This assessment requires experimental confirmation of the assumption made about the direct relationship between the $I_{D\,MIN}$, $I_{D\,MAX}$ currents and the activity of the input signals, since these currents are determined taking into account errors that reduce the direct dependence.

4 Experimental Comparative Assessment in Power-Oriented Checkability of Iterative Array and Bitwise Pipeline Multipliers

Experimental assessment of the lower power-oriented checkability of the circuit is performed by comparing its values for iterative array and bitwise pipeline multipliers according to the results of their simulation, which was performed in Quartus Prime CAD. When carrying out simulations on FPGA Intel Max 10 10M50DAF672I7G [45], designs of multipliers with a size of input operands n = 8, 16, 24, and 32 bits were implemented. The A_I activity of the input information signals was set in the range from 0% to 100% of the value of the clock signal with an increment of 12.5%. The frequency of the clock signal was set as the maximum possible for a specific multiplier project.

Iterative array multipliers were designed in Intel's FPGA Quartus Prime CAD based on an Intellectual Property Core (IP-Core) LPM_MULT of multiplier from the Library of Parameterized modules (LPM) that came with Quartus Prime. This IP-Core is implemented by CAD in the 9-bit multiplication blocks embedded in the FPGA Intel Max 10. The input and output user buffer registers were added to the LPM_MULT IP-Core (Fig. 1).

Bitwise pipeline multipliers were designed in Quartus Prime based on the circuit described in [30], (Fig. 2).

The Time-Quest Timing Analyzer utility [46] was used to set the clock signal values and adjust the temporal parameters of the functioning for multipliers projects. The maximum possible frequency for each of the projects was determined by Quartus Prime as a result of the compilation of projects.

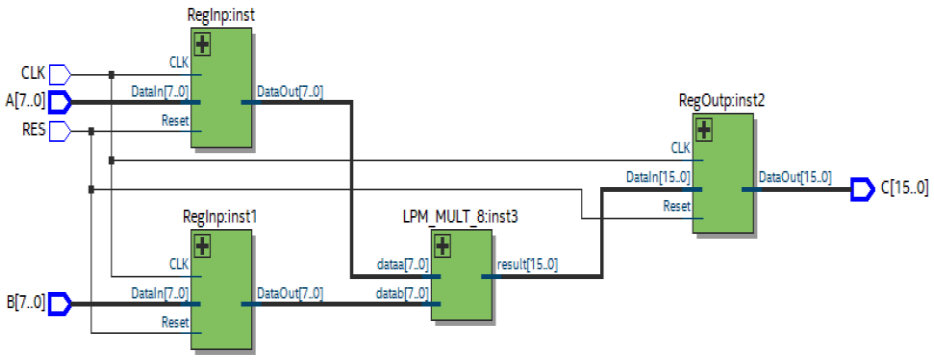

Fig. 1. An example of the project of an iterative array 8-bit multiplier

The Power-Play Power Analyzer utility was used to model the power consumption parameters of the multipliers. Before performing the simulation, it allows to set the parameters for calculating the activity of input and internal information signals.

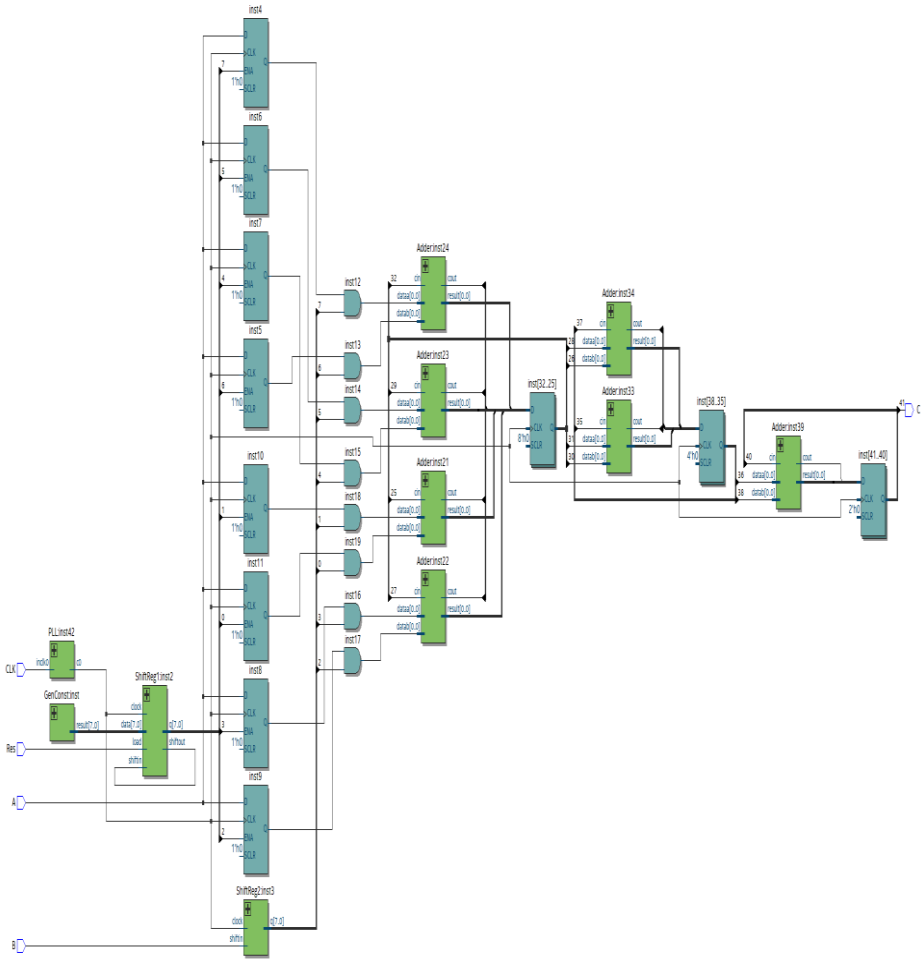

Fig. 2. An example of the project of a bitwise pipeline 8-bit multiplier

The simulation results, which are the values of currents I_T, I_S, I_D of the FPGA core, are given in Table 1 and 2, respectively, for the iterative array and bitwise pipeline multipliers with a size of n from 8 to 32 bits.

The maximum clock frequency obtained as a result of modeling for the iterative array multipliers is 250 MHz, 232 MHz, 111 MHz and 100 MHz, respectively, at 8-, 16-, 24- and 32-bit operands.

For bitwise pipeline multipliers, the maximum frequency is the same for all digits and is 400 MHz.

Tab. 1. Experiment results for iterative array multipliers

A_I, %	8 bit			16 bit			24 bit			32 bit		
	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	6.67	11.70	18.38	7.76	11.75	19.51	8.26	11.71	19.96	8.49	11.72	20.21
12.5	7.08	11.70	18.79	8.61	11.76	20.36	9.63	11.71	21.34	9.87	11.73	21.59
25	7.49	11.71	19.20	9.46	11.76	21.22	11.00	11.72	22.71	11.24	11.73	22.98
37.5	7.90	11.71	19.61	10.31	11.76	22.08	12.37	11.72	24.09	12.62	11.74	24.36
50	8.31	11.71	20.02	11.16	11.77	22.93	13.74	11.73	25.47	14.00	11.74	25.74
62.5	8.71	11.71	20.43	12.02	11.77	23.79	15.11	11.73	26.84	15.38	11.75	27.13
75	9.12	11.72	20.84	12.87	11.78	24.64	16.48	11.74	28.22	16.75	11.76	28.51
87.5	9.53	11.72	21.25	13.72	11.78	25.50	17.85	11.74	29.59	18.13	11.76	29.89
100	9.94	11.72	21.66	14.57	11.79	26.36	19.22	11.75	30.97	19.51	11.77	31.27

Tab. 2. Experiment results for pipeline multipliers

A_I, %	8 bit			16 bit			24 bit			32 bit		
	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	10.28	11.67	21.95	11.77	11.67	23.44	16.94	11.68	28.62	17.94	11.68	29.63
12.5	10.53	11.67	22.20	12.26	11.67	23.93	17.61	11.68	29.29	18.86	11.69	30.55
25	10.78	11.67	22.44	12.75	11.67	24.42	18.28	11.69	29.97	19.78	11.69	31.46
37.5	11.02	11.67	22.69	13.23	11.67	24.91	18.95	11.69	30.64	20.69	11.69	32.38
50	11.27	11.67	22.94	13.72	11.68	25.40	19.63	11.69	31.32	21.61	11.69	33.30
62.5	11.40	11.67	23.07	13.94	11.68	25.61	19.92	11.69	31.61	22.01	11.70	33.71
75	11.53	11.67	23.20	14.15	11.68	25.83	20.21	11.69	31.90	22.41	11.70	34.11
87.5	11.65	11.67	23.33	14.36	11.68	26.04	20.50	11.69	32.20	22.82	11.70	34.52
100	11.78	11.67	23.46	14.57	11.68	26.25	20.80	11.69	32.49	23.22	11.70	34.92

The results of experimental verification of the minimum value of the current $I_{D\,MIN}$ with zero input signal activity are presented in Table 3

Tab. 3. $I_{D\,MIN}$ current values for different input signal activity

A_I, %	Iterative array multipliers				Pipeline multipliers			
	8 bit	16 bit	24 bit	32 bit	8 bit	16 bit	24 bit	32 bit
0	5.92	6.98	7.47	7.69	9.44	10.89	15.93	16.91
12.5	6.32	7.81	8.80	9.04	9.68	11.37	16.59	17.80
25	6.72	8.64	10.14	10.37	9.93	11.85	17.24	18.70
37.5	7.12	9.46	11.47	11.72	10.16	12.32	17.89	19.59
50	7.52	10.29	12.81	13.06	10.40	12.79	18.55	20.49
62.5	7.91	11.13	14.15	14.41	10.53	13.01	18.84	20.87
75	8.31	11.96	15.48	15.74	10.66	13.21	19.12	21.26
87.5	8.71	12.79	16.82	17.09	10.78	13.42	19.40	21.66
100	9.11	13.62	18.15	18.43	10.90	13.62	19.70	22.05

The results of the checkability calculations according to the formula (4) for iterative array and bitwise pipeline multipliers are presented in Table 4.

Tab. 4. Power Consumption Checkability

A_I, %	Iterative array multipliers				Pipeline multipliers			
	8 bit	16 bit	24 bit	32 bit	8 bit	16 bit	24 bit	32 bit
0	79.74	81.70	82.51	82.81	84.88	86.12	88.77	89.11
12.5	75.46	74.14	71.42	71.87	82.97	82.83	85.50	84.89
25	71.62	67.85	62.97	63.53	81.15	79.78	82.46	81.06
37.5	68.16	62.55	56.30	56.88	79.46	77.01	79.63	77.59
50	65.01	58.02	50.91	51.49	77.79	74.37	76.95	74.37
62.5	62.21	54.06	46.46	47.04	76.94	73.24	75.86	73.05
75	59.57	50.64	42.73	43.32	76.11	72.19	74.80	71.78
87.5	57.16	47.63	39.55	40.12	75.37	71.18	73.77	70.52
100	54.93	44.95	36.81	37.36	74.57	70.19	72.74	69.33

The table shows growth of the checkability with increase in size n and decrease of the A_I activity. The bitwise multiplier surpasses matrix circuits in a checkability and reduces it to a lesser extent with growth of the A_I activity.

5 Conclusions

The role of checkability of circuits increases in safety-related systems, since it is a necessary condition for converting fault-tolerant solutions into fault-safe.

The logical form of checkability has received the dominant development in testing and on-line testing of digital circuits as structural, structurally functional, and dual-mode structurally functional checkability, with the deficit of which the problem of hidden faults arises that is inherent to safety-related systems in the case of traditional component design based on matrix structures.

A drastic reduction of matrix structures in bitwise pipelines significantly improves logical checkability but requires significant changes in the design of digital components.

Another problem of logical checkability is faults in chains of the common signals, such as clock signals. These faults can fix the digital circuit in a state that is identified by the logical checking as correct.

The limitations of logical checkability in solving problems of hidden faults and monitoring of common signals stimulate the search for new forms of checkability.

The success of green technologies in FPGA design has created the conditions for the development of power-oriented checkability, which allows to detect faults in chains of the common signals by reducing the dynamic component of energy consumption.

Analytical evaluation of power-oriented checkability and experimental studies showed its increase from 54.3% to 79.7% in case of a decrease of the activity of input

signals from 100% to zero in an 8-bit iterative array multiplier and increase from 37.4% to 82.8 % for a 32-bit multiplier.

Bitwise pipelines demonstrate higher power-oriented checkability, which, under the same conditions, rises from 74.6% to 84.9% and from 69.3% to 89.1%.

Thus, power-oriented checkability significantly complements the possibilities of logical checkability for both traditional matrix circuits and promising bitwise pipelines.

References

1. Core Knowledge on Instrumentation and Control Systems in Nuclear Power Plants, Technical Reports, IAEA Nuclear energy series no. NP-T-3.12–141 p., International atomic energy agency Vienna (2011).
2. Yastrebenetsky, M. (ed): NPP I&Cs: Problems of Safety, Ukraine, Kyiv: Technika (2004).
3. 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).
4. 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).
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. Automation and Remote Control, vol. 63, no. 6, 991–1003 (2002).
6. 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).
7. IEC 61508-1:2010. Functional Safety of Electrical / Electronic / Programmable Electronic Safety Related Systems – Part 1: General requirements. Geneva: International Electrotechnical Commission (2010).
8. Bakhmach, E., Kharchenko, V., Siora, A., Sklyar, V., Tokarev, V.: Design and Qualification of I&C Systems on the Basis of FPGA Technologies. In: Proceedings of 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT 2010) , pp. 916–924. Las Vegas, Nevada (2010).
9. 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).
10. Hahanov, V., Litvinova, E., Obrizan, V., Gharibi, W: Embedded method of SoC diagnosis. Elektronika in Elektrotechn, no 8, pp. 3–8 (2008).
11. 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
12. 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).

13.Romankevich, V.A.: Self-testing of multiprocessor systems with regular diagnostic connections. Automation and Remote Control, vol. 78, Issue 2, 289–299 (2017).
14.Nicolaidis, M., Zorian, Y.: On-Line Testing for VLSI – A Compendium of Approaches. In: Journal of Electronic Testing: Theory and Application, vol. 12, issue 1–2, pp. 7–20 (1998).
15.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, NJ, USA (1999).
16. 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.
17.Department of Energy, DOE Guideline Root Cause Analysis Guidance Document, Office of Nuclear Energy and Office of Nuclear Safety Policy and Standards, Department of Energy, Washington DC, USA, 1992. http://tis.eh.doe.gov/techstds/standard/nst1004/nst1004.pdf.
18.Kulanov, V., Kharchenko, V., Perepelitsyn, A.: Parameterized IP Infrastructures for fault-tolerant FPGA-based systems: Development, assessment, case-study. In: IEEE East-West Design and Test Symposium EWDTS'10, pp. 322–325. St. Petersburg, Russia (2010). DOI: 10.1109/EWDTS.2010.5742075.
19.Li, H.F.: A structural study of parallel pipelined systems. In: PhD Dissertation, Univ. of California., Berkeley (1975).
20.Cadenas, O., Megson, G.: A clocking technique for FPGA pipelined designs. Journal of System Architecture, no. 50, 687–696 (2004).
21.Scott J.A., Preckshot G.G., Gallagher J.M. Using Commercial-Off-The-Shelf (COTS) Software in High-Consequence Safety Systems, Lawrence Livermore National Laboratory, UCRL – 122246 (1995).
22.Correia, M., Veríssimo, P., Neves, N. F.: The design of a COTS real-time distributed security kernel. In: Fourth European Dependable Computing Conference, pp. 234–252. Toulouse, France (2002).
23.Kondratenko, Y., Gordienko, E.: Implementation of the neural networks for adaptive control system on FPGA. In: Annals of DAAAM for 2012 & 23th Int. DAAAM Symp. "Intelligent Manufacturing and Automation, vol. 23, no. 1, B. Katalinic (ed), pp. 389–392. DAAAM International, Vienna, Austria (2012).
24.Tyurin, S.F., Grekov, A.V., Gromov, O.A.: The principle of recovery logic FPGA for critical applications by adapting to failures of logic elements. World Applied Sciences Journal, 328–332 (2013). DOI: 10.5829/idosi.wasj.2013.26.03.13474.
25.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).
26.Yakovlev, A.: Energy-modulated computing. In: Proceedings of Design, Automation and Test in Europe Conference and Exhibition 2011 (DATE 2011), pp. 1–6. Grenoble, France (2011).
27.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.
28.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: 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 233–237. Berlin, Germany (2013). DOI: 10.1109/IDAACS.2013.6662656.
29.Palagin, A., Opanasenko, V.: The implementation of extended arithmetic's on FPGA-based structures. In: 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, vol. 2, pp. 1014–1019. Bucharest, Romania (2017). DOI: 10.1109/IDAACS.2017.8095239.

30. 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.
31. Cyclone II Architecture. Cyclone II Device Handbook Version 3.1.—Altera Corporation (2007), http://www.altera.com/literature/hb/cyc2/cyc2_cii51002.pdf, last accessed 2019/03/20.
32. 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).
33. Park, H.: Truncated Multiplications and Divisions for the Negative Two's Complement Number System. In: Ph.D. Dissertation. The University of Texas at Austin, Austin, USA (2007).
34. 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).
35. ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic (1985).
36. IEEE Std 754™-2008 (Revision of IEEE Std 754-1985) IEEE Standard for Floating-Point Arithmetic. IEEE 3 Park Avenue New York, NY 10016–5997, USA (2008).
37. Cyber Physical Computing for IoT-driven Services. Vladimir Hahanov, Eugenia Litvinova, Svetlana Chumachenko. Springer (2017).
38. Maevsky, D., Bojko, A., Maevskaya, E., Vinakov, O., Shapa, L.: Internet of Things: Hierarchy of Smart Systems. In: 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, vol. 2, pp. 821–827, Bucharest, Romania (2017).
39. 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.
40. Drozd, A.V., Lobachev, M.V.: Efficient On-line Testing Method for Floating-Point Adder. In: Design, Automation and Test in Europe. Conference and Exhibition 2001 (DATE 2001), pp. 307–311. Munich, Germany (2001). DOI: 10.1109/DATE.2001.915042.
41. 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.
42. 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.
43. 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.
44. 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.
45. 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.
46. 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.